Practical Electronics

The UK's premier electronics and computing maker magazine

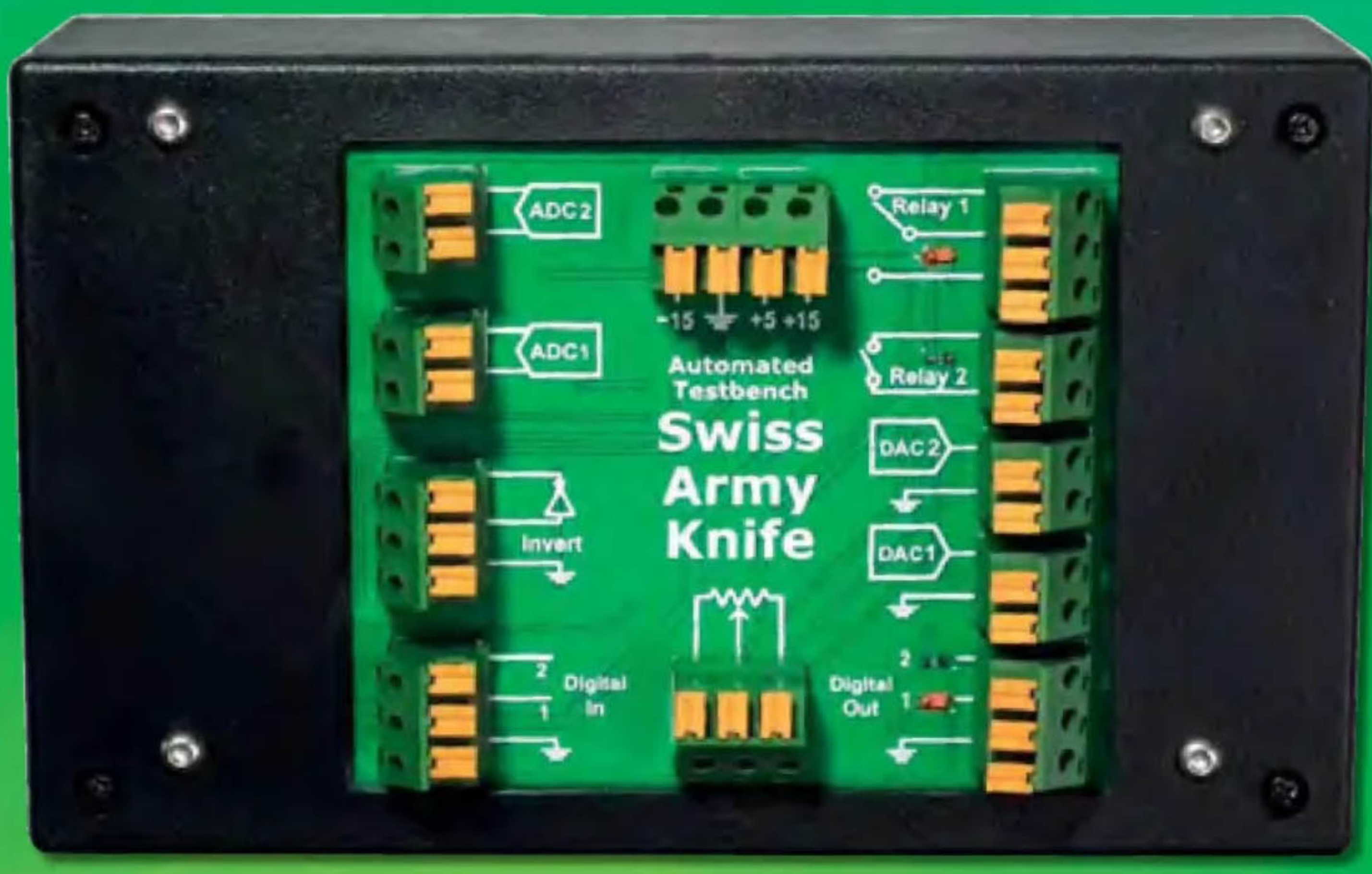
Circuit Surgery
LTspice 24 update and Frequency Response Analysis

Teach-In 2024
Digital input and output with the ESP32 SoC microcontroller

Audio Out
Designing discrete buffers

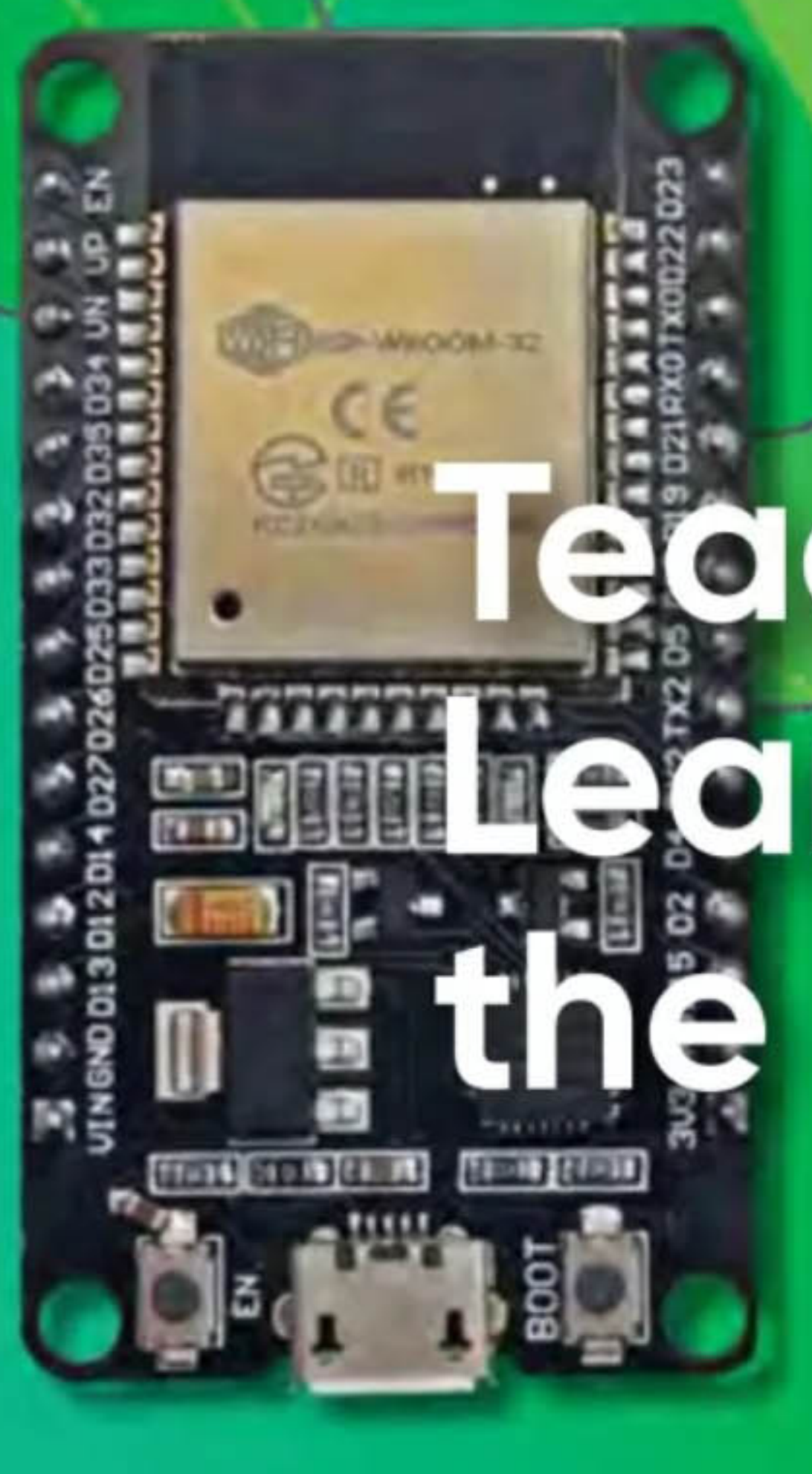
AD9834 DDS
SIGNAL
GENERATOR
ON
FREQ

Basic RF Signal Generator

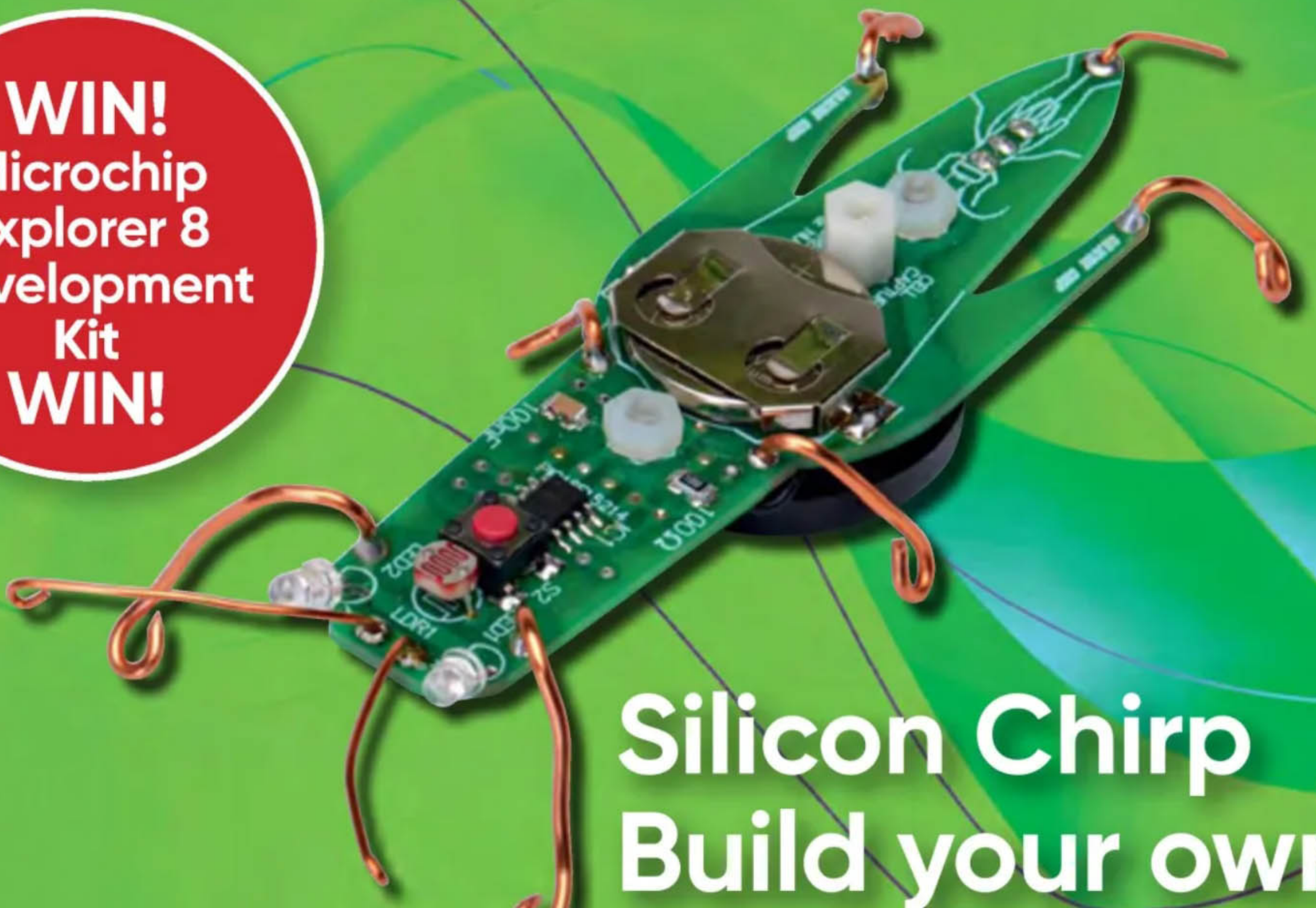

ADC2
ADC1
Invert
Digital In
Automated Testbench
Swiss Army Knife
Relay 1
Relay 2
DAC2
DAC1
Digital Out

'Swiss Army Knife' Automated Test Bench

WIN!
Microchip Explorer 8 Development Kit
WIN!



Teach-In 2024 Learn to use the ESP32



Silicon Chirp Build your own pet cricket

PLUS!

Techno Talk – Wait! What? Really?

Cool Beans – Arduino: expanding the 7-segment display

Net Work – Label printers, smart home devices and SMRs

Apr 2024 £5.99

9 772632 573030 04

www.electronpublishing.com | @practicalelec | practicalelectronics

15:34
157 bpm
2.31km
54%
15:01
Average Duration 07:45
Average Duration 00:45 AM
Average Training Quality 75%

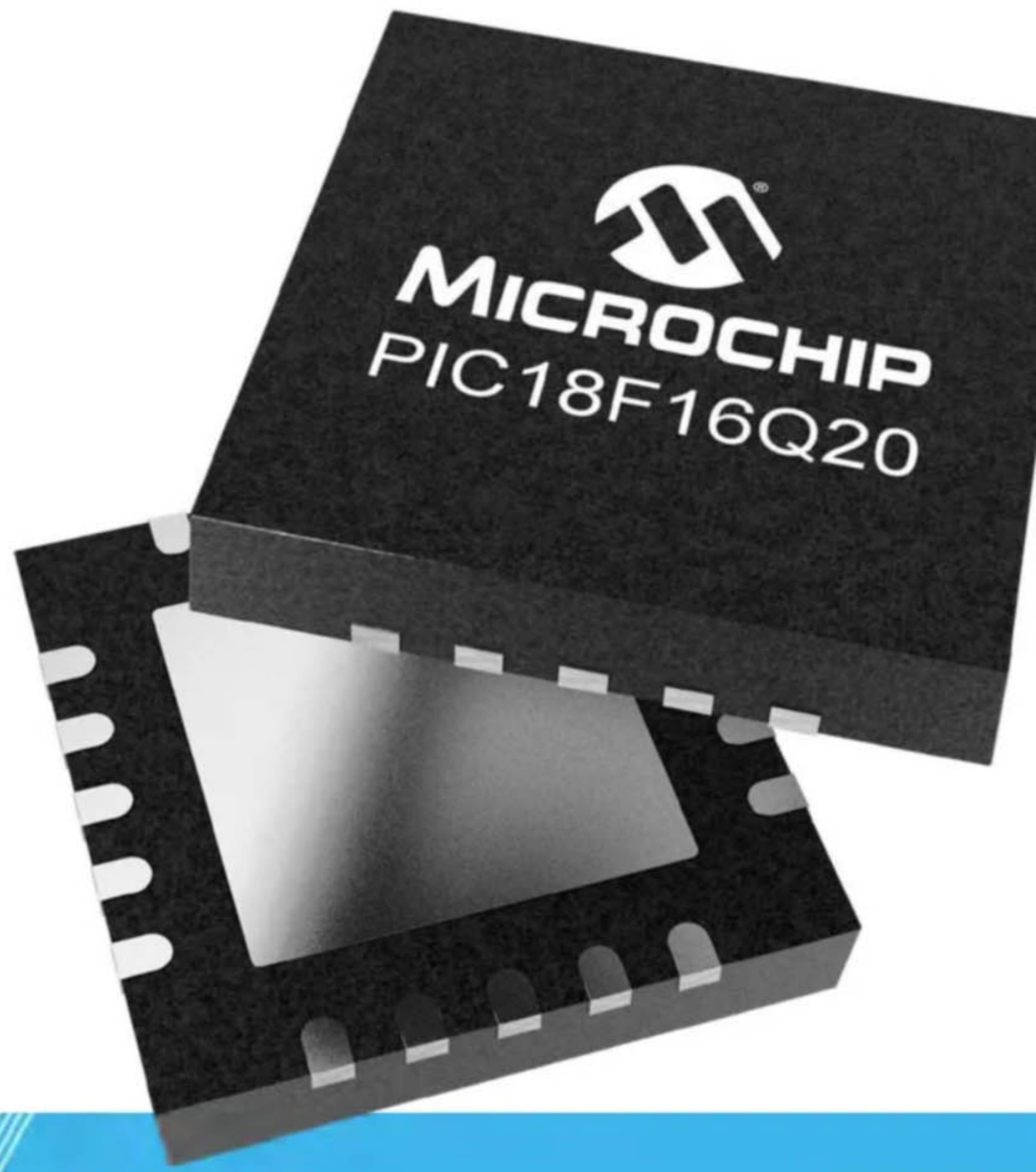
Flexible Communication with I3C® and I2C Compatibility

With the Versatile PIC18-Q20 Family of MCUs

The PIC18-Q20 family of microcontrollers (MCUs) offers configurable peripherals, advanced communication interfaces and supports multiple voltage domains for embedded systems containing several sensors. These MCUs are equipped with I3C® modules that have a fast communication rate, a high-speed 10-bit Analog-to-Digital Converter with Computation (ADCC), capacitive touch sensing and an 8-bit signal routing port to interconnect digital peripherals. This product family easily interfaces across multiple voltage domains without external components, supporting 1V operation for I3C communication. The MCUs are well-suited as a I3C-to-I2C communication bridge or a companion MCU for larger IoT systems to perform housekeeping functions for an extensive range of space-sensitive applications and markets including IoT, computing, consumer, medical, wearables, touch sensing and memory management applications.

Key Features

- Up to two I3C interface (12.5 MHz, 1.0–3.6V operation)
- I2C, SPI and UART communication interfaces
- Up to two multiple voltage domains



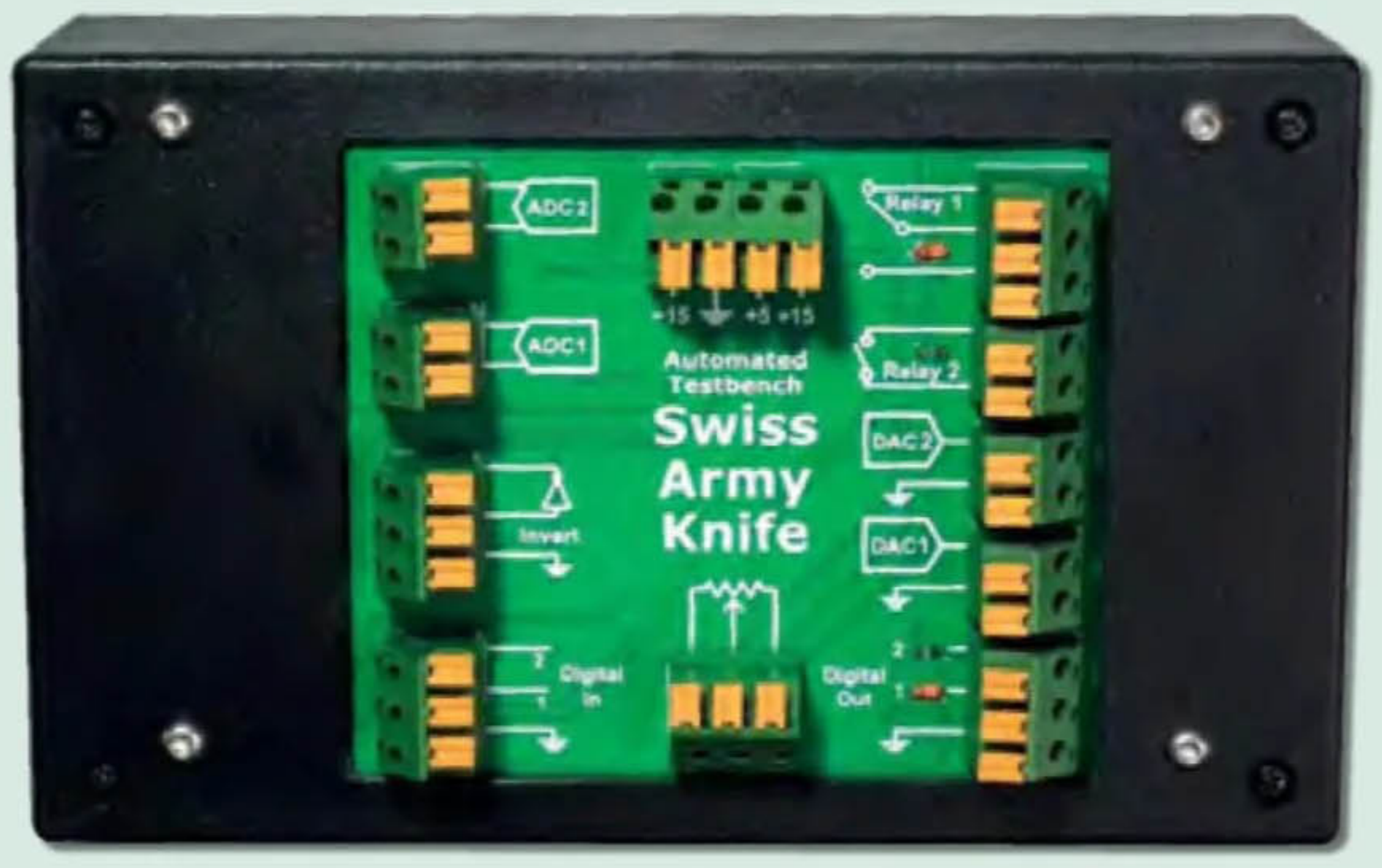
microchip.com/q20

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2024 Microchip Technology Inc. All rights reserved.

Practical Electronics

Volume 53. No. 4
April 2024
ISSN 2632 573X

Contents



AD9834 DDS SIGNAL GENERATOR
ON
FREQ

Projects and Circuits

Basic RF Signal Generator by Charles Kosina 16
Our AM/FM/DDS Signal Generator design from May 2023 is a great piece of test equipment, but overkill for many tasks. If you only need a basic test signal from 10Hz to 25MHz, then this Generator is just right – compact and cheap to build.

Automated Test Bench by Richard Palmer 24
This 'Swiss Army Knife' provides test voltages/signals, varies a resistance, switches a component in/out of circuit and even measures some voltages. It uses automation software, testing automatically and records input and output values for analysis.

Silicon Chirp by John Clarke 34
This pet cricket will keep you company; it only needs to be fed occasionally and won't run away. Keep it for yourself or play a prank on a family member or friend by hiding it in their room. When they switch the lights off, they will get a bit of a surprise!

Series, Features and Columns

Techno Talk by Max The Magnificent 8
Wait! What? Really?

Net Work by Alan Winstanley 10
Reports on a bargain label printer from China, plus smart home products by TP-Link's Tapo are examined. There's also news on SMR power generation.

NEW SERIES!
Teach-In 2024 – Learn electronics with the ESP32 by Mike Tooley 40
Part 2 – Digital input and output

Circuit Surgery by Ian Bell 50
LTspice 24 and Frequency Response Analysis – Part 2

Max's Cool Beans by Max The Magnificent 56
Arduino Bootcamp – Part 16

Audio Out by Jake Rothman 62
Back to the buffers – Part 3

Regulars and Services

Wireless for the Warrior 2
Subscribe to Practical Electronics and save money 4
NEW! Practical Electronics back issues DOWNLOADS – 2022 now available! 6
Reader services – Editorial and Advertising Departments 7
Editorial 7
Welcome to the future
Exclusive Microchip reader offer 9
Win a Microchip Explorer 8 Development Kit
Teach-In bundle – what a bargain! 23
PE Teach-In 9 39
PE Teach-In 8 61
PCBs for Practical Electronics projects
Practical Electronics PCB Service 68
Classified ads and Advertiser index 71
Next month! – highlights of our next issue of Practical Electronics 72

Made in the UK.

Written in Britain, Australia, the US and Ireland.

Read everywhere.

© Electron Publishing Limited 2024

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The May 2024 issue of *Practical Electronics* will be published on Thursday, 4 April 2024 – see page 72.

WIRELESS FOR THE WARRIOR

by LOUIS MEULSTEE

THE DEFINITIVE TECHNICAL HISTORY OF RADIO COMMUNICATION EQUIPMENT IN THE BRITISH ARMY

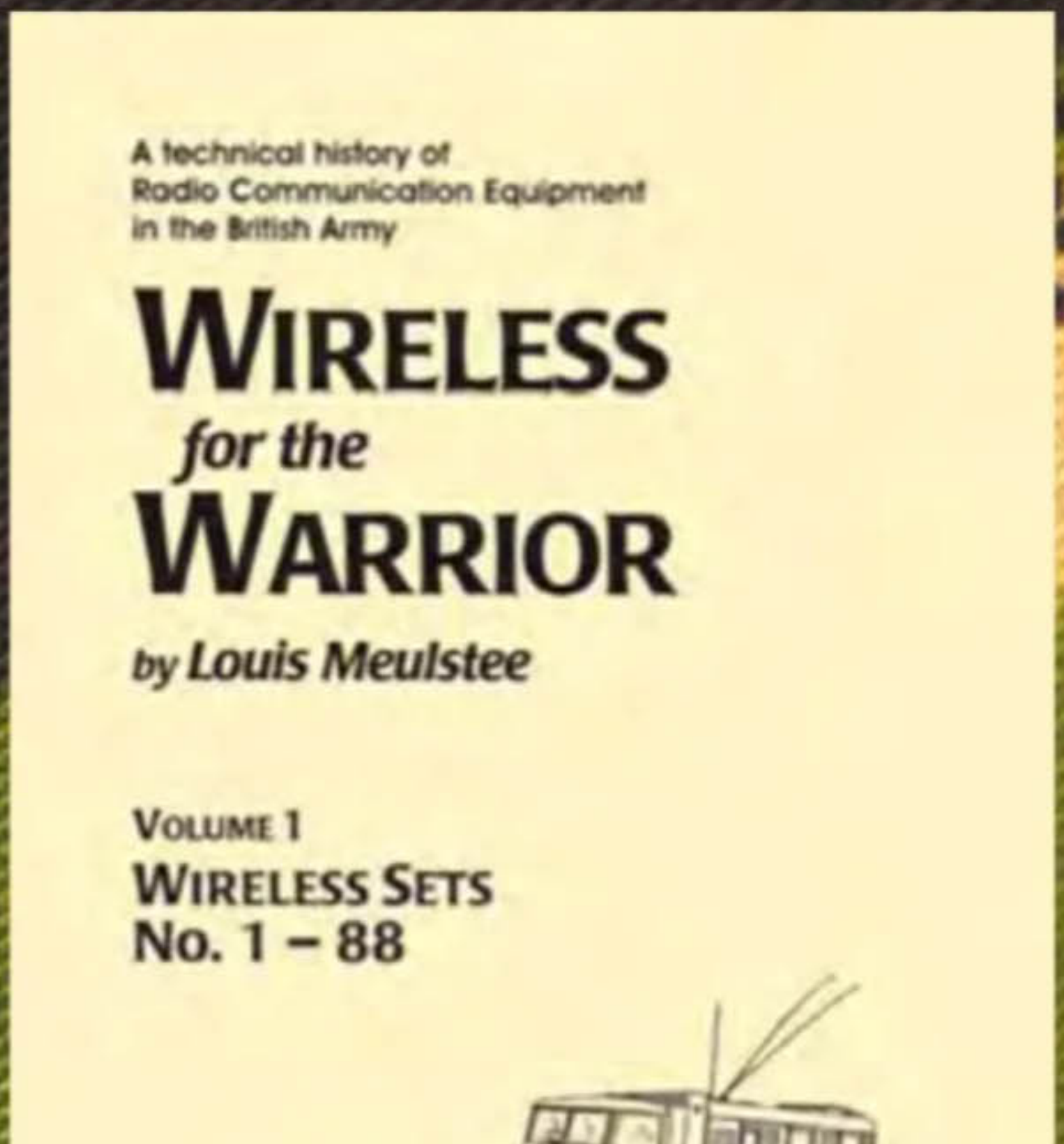
The *Wireless for the Warrior* books are a source of reference for the history and development of radio communication equipment used by the British Army from the very early days of wireless up to the 1960s.

The books are very detailed and include circuit diagrams, technical specifications and alignment data, technical development history, complete station lists and vehicle fitting instructions.

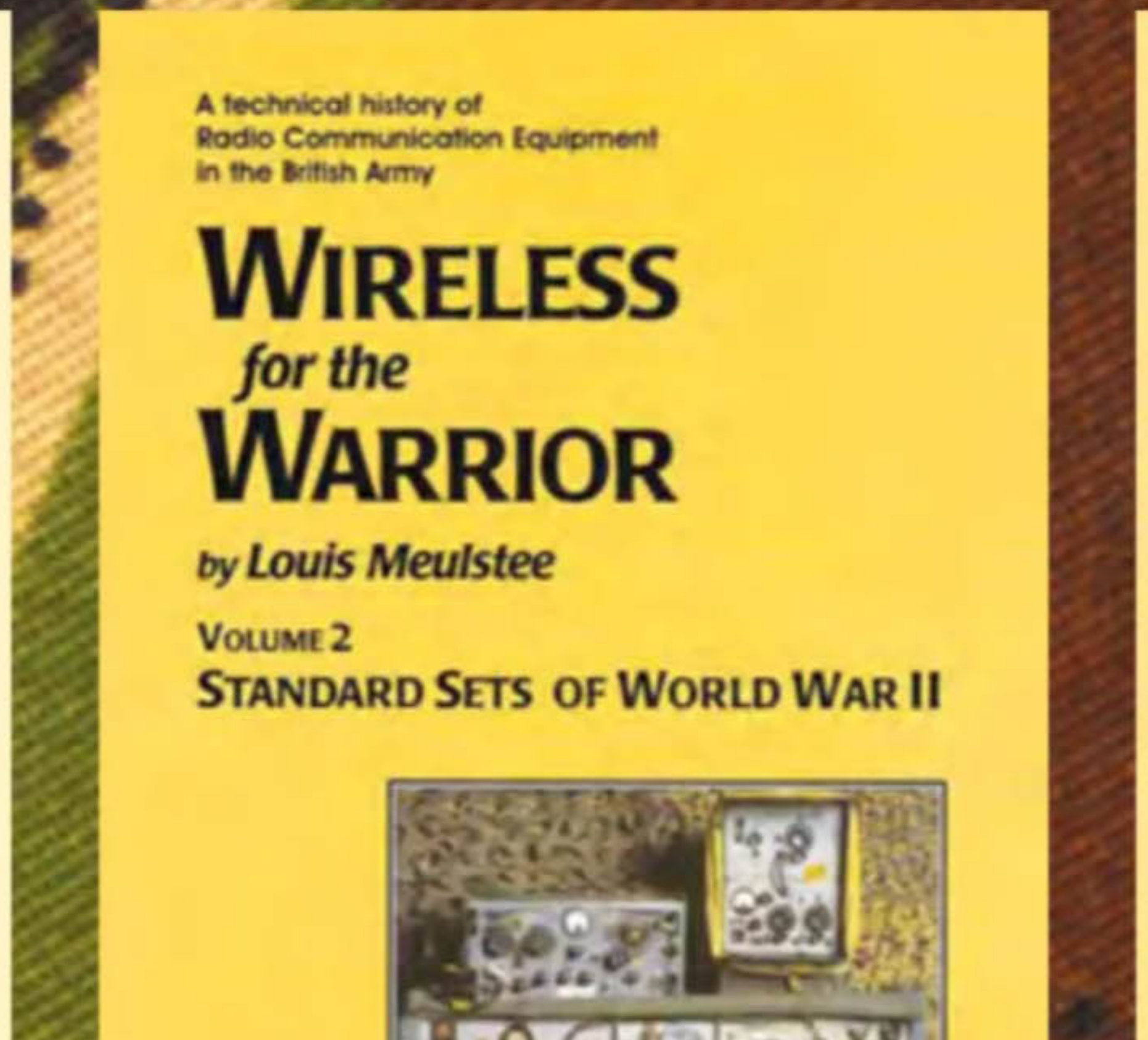
***Volume 1* and *Volume 2* cover transmitters and transceivers used between 1932-1948. An era that starts with positive steps taken to formulate and develop a new series of wireless sets that offered great improvements over obsolete World War I pattern equipment. The other end of this timeframe saw the introduction of VHF FM and hermetically sealed equipment.**

***Volume 3* covers army receivers from 1932 to the late 1960s. The book not only describes receivers specifically designed for the British Army, but also the Royal Navy and RAF. Also covered: special receivers, direction finding receivers, Canadian and Australian Army receivers, commercial receivers adopted by the Army, and Army Welfare broadcast receivers.**

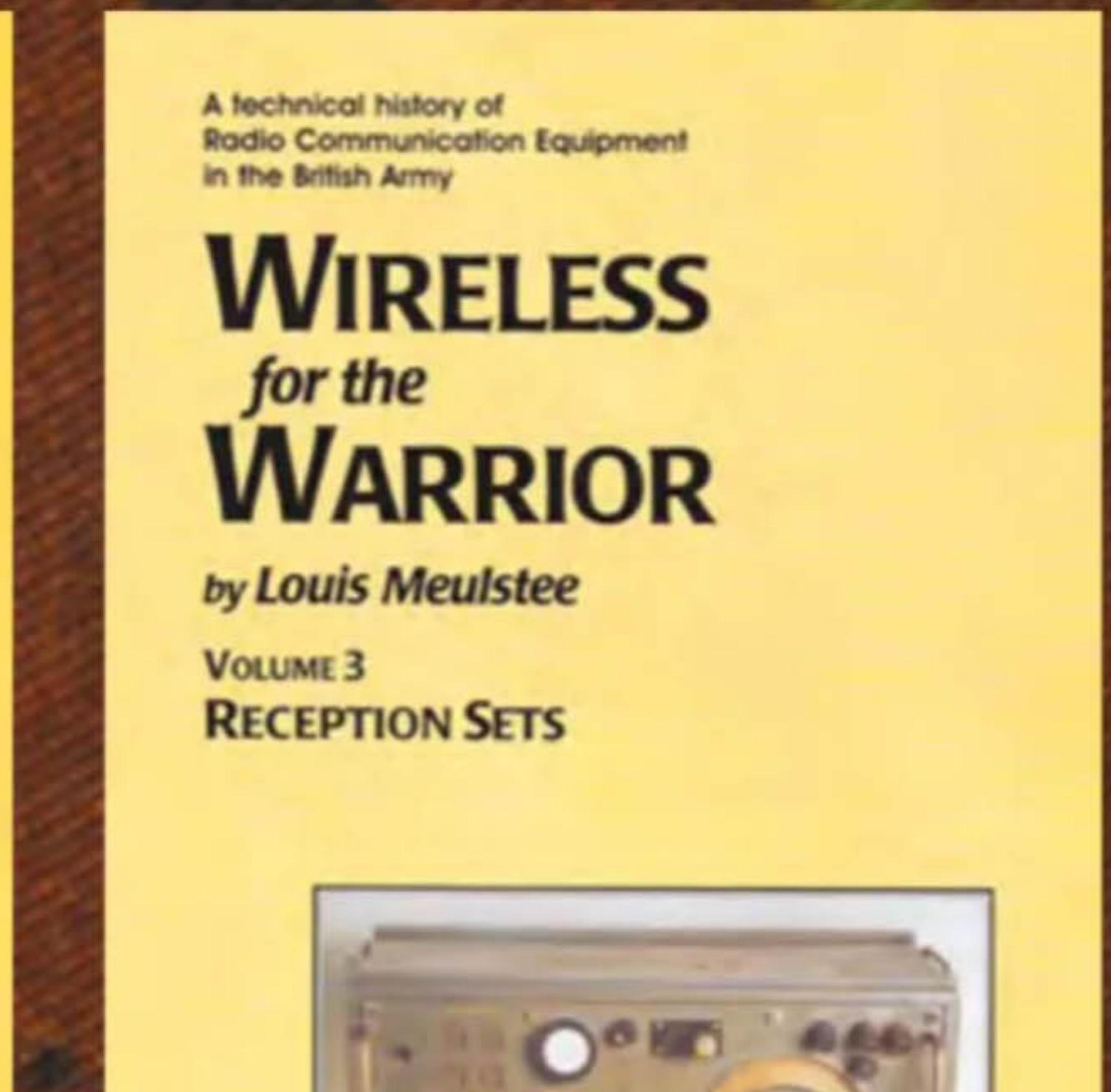
***Volume 4* covers clandestine, agent or 'spy' radio equipment, sets which were used by special forces, partisans, resistance, 'stay behind' organisations, Australian Coast Watchers and the diplomatic service. Plus, selected associated power sources, RDF and intercept receivers, bugs and radar beacons.**



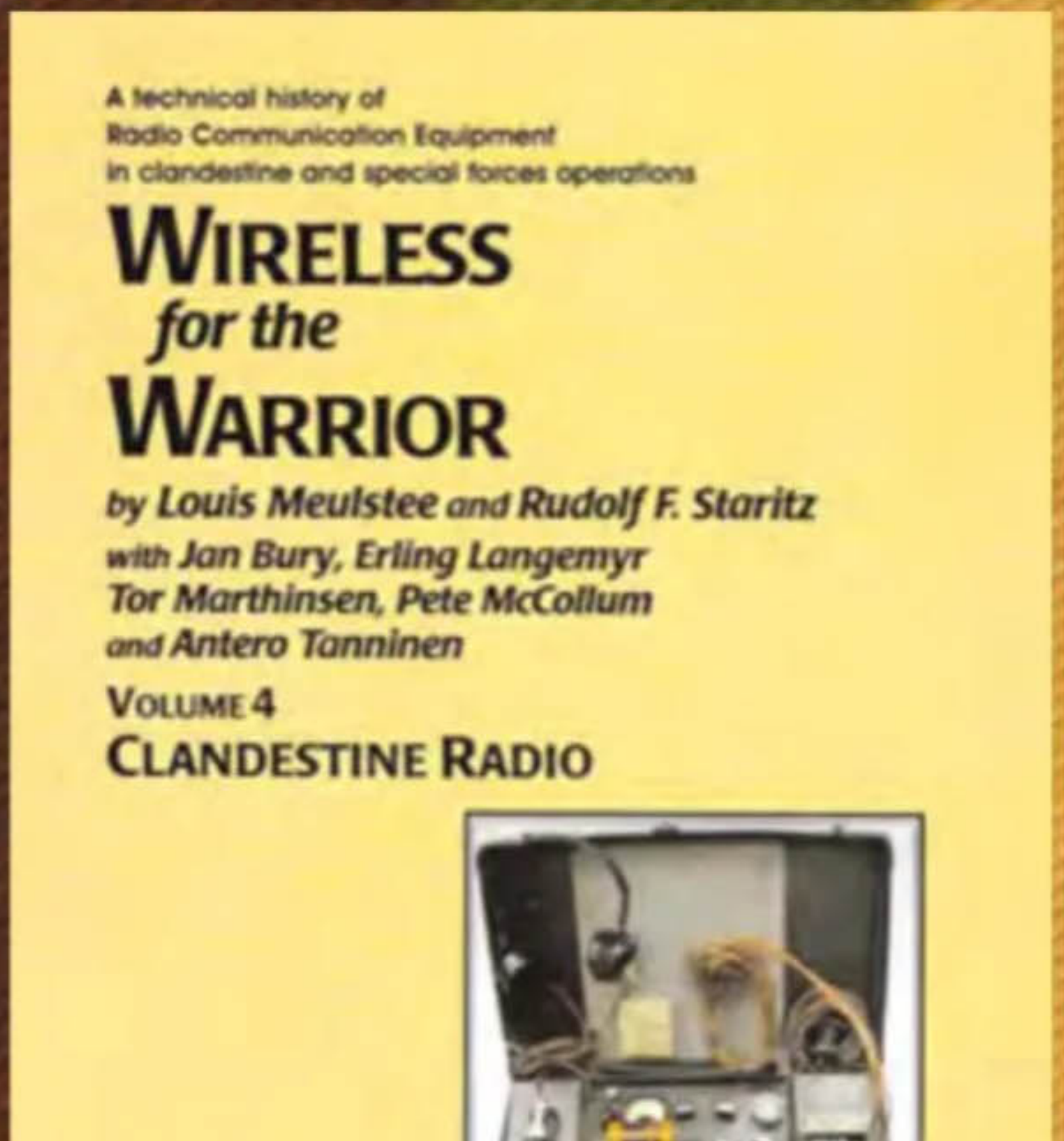
A technical history of
Radio Communication Equipment
in the British Army
WIRELESS
for the
WARRIOR
by Louis Meulstee
VOLUME 1
WIRELESS SETS
No. 1 – 88



A technical history of
Radio Communication Equipment
in the British Army
WIRELESS
for the
WARRIOR
by Louis Meulstee
VOLUME 2
STANDARD SETS OF WORLD WAR II



A technical history of
Radio Communication Equipment
in the British Army
WIRELESS
for the
WARRIOR
by Louis Meulstee
VOLUME 3
RECEPTION SETS



A technical history of
Radio Communication Equipment
in clandestine and special forces operations
WIRELESS
for the
WARRIOR
by Louis Meulstee and Rudolf F. Staritz
with Jan Bury, Erling Langemyr
Tor Marthinsen, Pete McCollum
and Antero Tanninen
VOLUME 4
CLANDESTINE RADIO

ORDER YOURS TODAY!
JUST CALL 01202 880299 OR VISIT www.electronpublishing.com

QUASAR electronics

Quasar Electronics Limited
PO Box 6935, Bishops Stortford
CM23 4WP, United Kingdom

Tel: 01279 467799
E-mail: sales@quasarelectronics.co.uk
Web: quasarelectronics.co.uk

All prices include 20% VAT. Free UK mainland delivery on orders over £60. Postage & Packing Options (Up to 1Kg gross weight): **UK Standard 2-5 Day Delivery -** £4.95 : **UK Mainland Next Day Delivery -** £9.95 : Please order online if you reside outside the UK (our website will calculate postage for you).
Payment: We accept all major credit/debit cards. Make UK cheques/PO's payable to Quasar Electronics Limited and include P&P detailed above.
!! Order online for reduced price postage and fast despatch !!
Please visit our online shop now for full details of over 1000 electronic kits, projects, modules and publications. Discounts for bulk quantities.

QuasarElectronics.co.uk

Solutions for Home, Education & Industry Since 1993

cebek
WHADDA EXCITING ELECTRONICS
velleman

Great Brands - Official Main Dealer

Electronic Kits & Modules

We have a massive selection of self-assembly electronic kits and pre-assembled modules. Please see the full range on our website or call for details.

LED Buddy / LED Tester Kit

Hold any type of LED to the contact pads to see it's polarity, forward voltage & the recommended series resistor value. Adjustable target current & forward voltage. Great design aid.
Order Code: WSMI198 - ***£16.66***

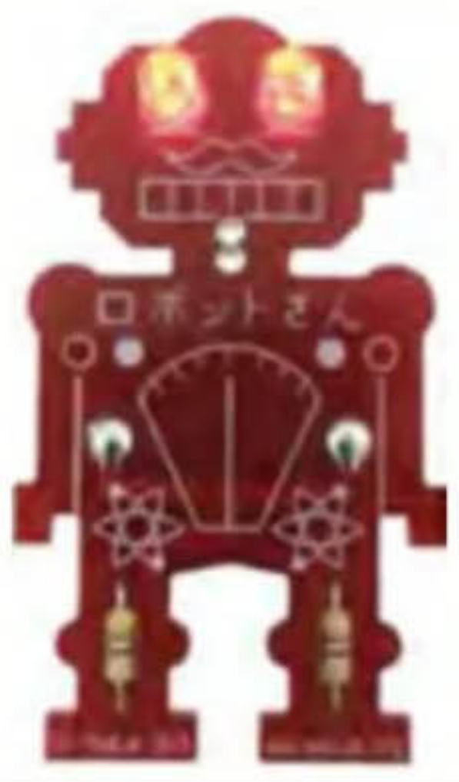
12-in-1 Solar Hydraulic Construction Kit

Solar & hydraulic powered robot can be transformed into twelve different animals and mechanical robots (monkey, T-Rex, scorpion, excavator, etc). Moves easily. Provides great interaction with kids. Teaches the benefit of alternative energy. Aged 14+.
Order Code: KSR17 - ***£29.95.***

Stereo Ultrasonic Bat Detector Kit

Educational & fun kit converts high frequency sounds (20 - 90kHz) normally imperceptible to humans like bat signals into audible noise. Can also help detect failures in machines, engines, etc. Stereo feature adds the possibility to pinpoint the source. Frequency range 20-90kHz. 3.5mm jack output. 129x60x40mm. Requires 3x AA batteries and stereo headphones (not included).
Order Code: WSAK8118 - ***£20.39***

NEW! Mr Robot Kit



A fun, wearable electronic gadget that you can pin to your clothes. Easy to solder beginners kit. Two flashing bicolour LEDs resemble the eyes. 65 x 40 x 15 mm. CR2032 battery powered (not included).
Order Code: WSL108 - 6.68

LED Electronic Dice Kit

Still popular! A must-build soldering kit for all beginners. Dice slowly rolls to stop on a random number when the push button is released. 9V battery (not included). Ideal for educational courses. Wide selection of component types.
Order Code: MK109 - ***£7.08***

3 Channel RGB LED Light Organ Kit

Sens
Low R
Mid G
High B
velleman R-G-B LED light organ

3 outputs react to different sound frequencies. On-board microphone picks up surrounding sound or music and drives the low, mid & high frequency outputs. Connect 12-24Vdc RGB or separate colour LED strips or LEDs (not included). Master & separate channel sensitivity adjustment. 1.25 A max. per channel. Panel mountable facia 127 x 44mm. Terminal block or jack power supply connection. Control knobs included.
Order Code: WSL209 - ***£14.10***

Signal Generator Kit

Sine wave, triangle, square wave and integrator (selectable through jumper) with 0 - 100Vrms adjustable output level. 1kHz (approx.) fixed signal frequency. 9V battery (not included).
Order Code: WSAH105 - ***£6.12***

Bidirectional DC Motor Speed Controller

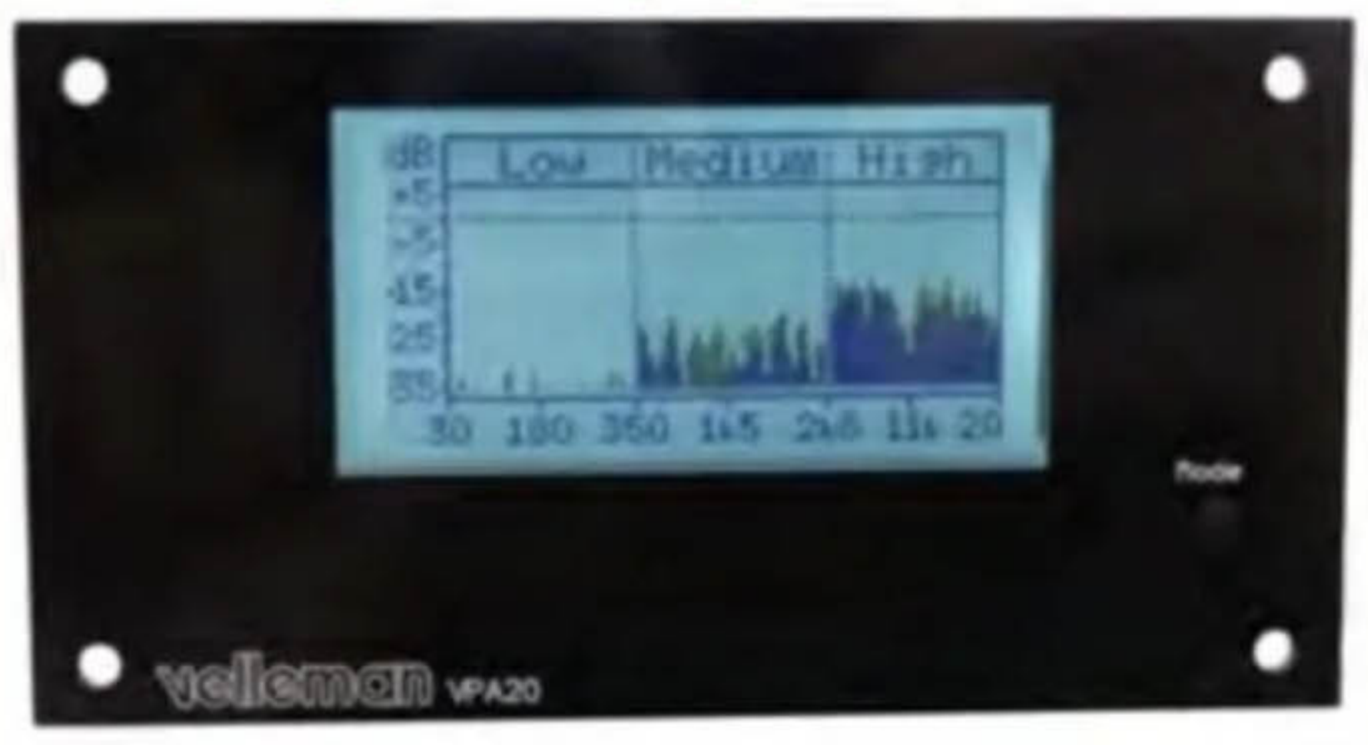
Control the speed of most common DC motors (rated up to 28Vdc/5A) from fully OFF to fully ON in **both directions**. Single potentiometer controls speed & direction. Screw terminal block connectors. PCB: 90x42mm. **Not suitable for use with lead acid batteries!**
Kit Order Code: 3166KT - ***£19.96***
Assembled Order Code: AS3166 - ***£29.95***

Card Sales & Enquiries 01279 467799

Digitally Controlled FM Radio Kit

Build your own modern, high quality FM receiver project with excellent sensitivity powered by a simple 9V PP3 battery (not included. Auto-seeking button. 4 station presets. Volume control. Excellent learning project for schools and colleges.
Order Code: WSAH194 - ***£20.39***

Audio Analyser Display Kit

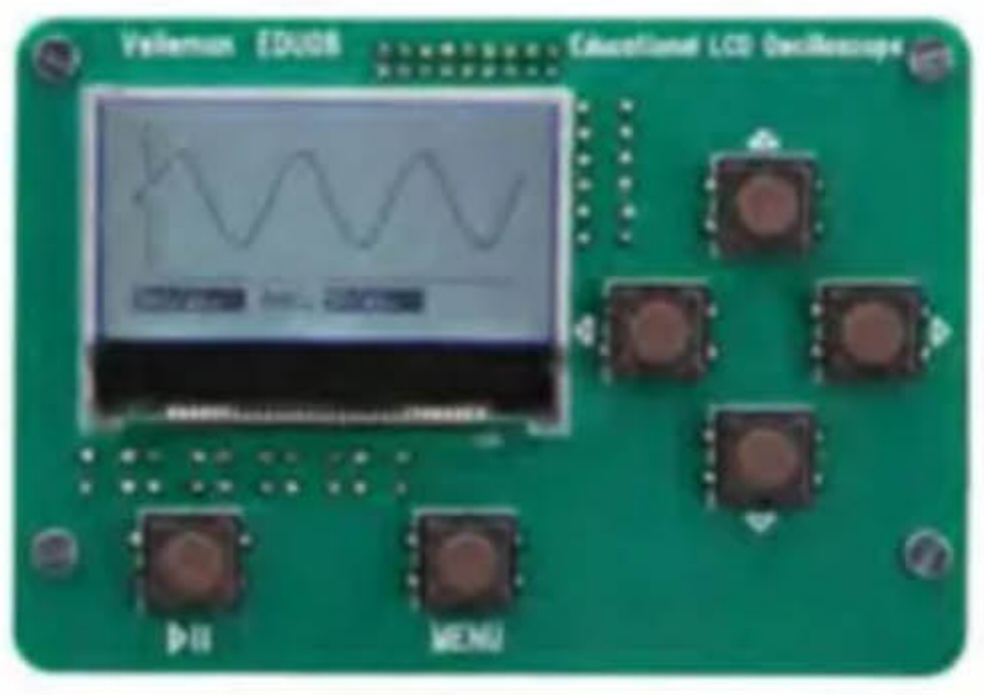


Small, compact LCD display, ideal for panel mounting. Give your homemade audio gear a high-tech look. Upgrade existing equipment. Provides Peak Power, RMS Power, Mean dB, Peak dB, Linear Audio Spectrum And 1/3 Octave Audio Spectrum. Auto / Manual range selection. Peak-hold function. Speaker impedance selection.
Order Code: K8098 - ***£38.39***

Electronic Component Tester Kit

Build your own versatile component tester. Shows value and pin layout information for resistors (0.1 Ohm resolution, max. 50M Ohm), coils (0.01mH - 20H), capacitors (28p - 100mF), diodes, BJT, JFET, E-IGBT, D-IGBT, E-MOS & D-MOS.
Order Code: WSMI8115 - ***£44.15***

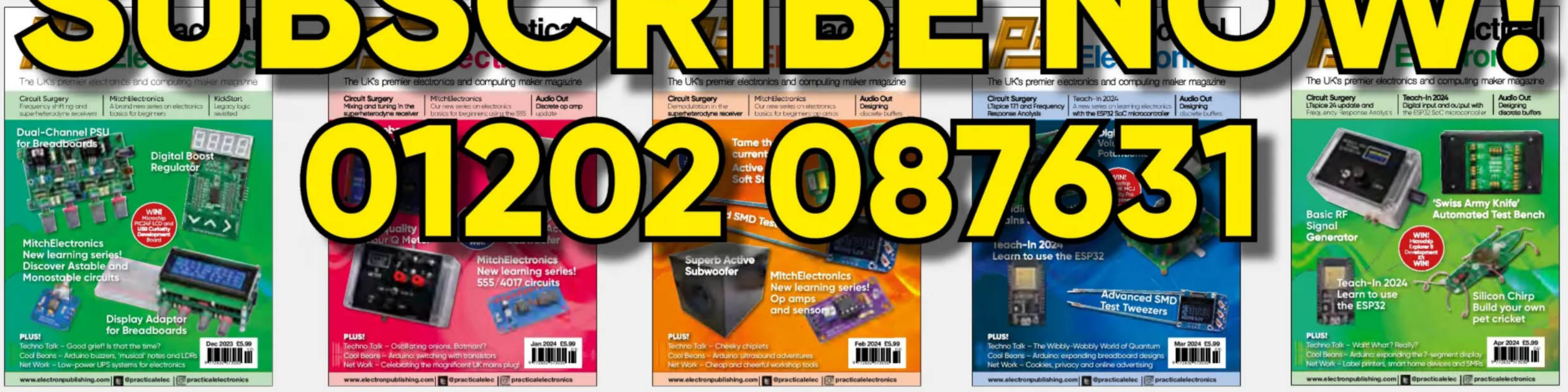
LCD Oscilloscope Educational Kit



Build your own LCD oscilloscope with this exciting new kit. Learn how to read signals. See the electronic signals you learn about displayed on your own LCD oscilloscope. Despite the low cost, this oscilloscope kit has a lot of features found on expensive units like signal markers, frequency, dB, true RMS readouts and more. A powerful auto-setup function will get you going in a flash!
Order Code: WSEDU08 - ***£48.54***

Practical Electronics

UK readers SAVE £1 on every issue



SUBSCRIBE NOW!

01202 087631

Take out a one-year subscription and save more than £10 over the year.
Even better – save over £1 per issue if you subscribe for two years – a total saving of £26.

Overseas rates represent exceptional value

You also:
- **Avoid any cover price increase for the duration of your subscription**
- **Get your magazine delivered to your door each month**
- **Guarantee a copy, even if your local newsagent sells out**

Order by:
- **Phone or post with a cheque, postal order or credit card**
- **Or order online at: www.electronpublishing.com**

Practical Electronics

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

UK: 6-months £33.99, 12-months £59.99, 24-months £114.99

Europe Airmail: 6-months £38.99, 12-months £69.99, 24-months £129.99

Rest Of The World Airmail: 6-months £46.99, 12-months £84.99, 24-months £164.99

Cheques (in **£ sterling only**) payable to *Practical Electronics* and sent to: Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

Also via our online shop at**: www.electronpublishing.com**

Subscriptions start with the next available issue. We accept MasterCard or Visa.
(For past issues see the Back Issues page.)

DIGITAL SUBSCRIPTIONS

Electronic subscriptions are available from £29.99 for 12 months, more details at: **www.electronpublishing.com**

SUBSCRIPTION ORDER FORM

☐ 6 Months: UK £33.99, Europe £38.99 (Airmail), Rest of the World £46.99 (Airmail)

☐ 1 Year: UK £59.99, Europe £69.99 (Airmail), Rest of the World £84.99 (Airmail)

☐ 2 Years: UK £114.99, Europe £129.99 (Airmail), Rest of the World £164.99 (Airmail)

To: **Practical Electronics Subscriptions**
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

I enclose payment of £ ....
(cheque/PO in £ sterling only)
payable to *Practical Electronics*

☐ Please charge my Visa/Mastercard

My card number is: ....
Please print clearly, and check that you have the number correct

Card Security Code .... Valid From Date.......................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ....

Name ....

Address ....

Post code .... Tel. ....

Email ....

Subscriptions start with the next available issue.

Subscribe to SILICON CHIP

SILICON CHIP
MAY 2023
PROJECTS:
SONGBIRD
TRALIA'S
ERN TIONAL AIRSHOW
HE NE EST TECH IN AVALON AIRPORT

Australia's top electronics magazine

Silicon Chip is one of the best DIY electronics magazines in the world. Each month is filled with a variety of projects that you can build yourself, along with features on a wide range of topics from in-depth electronics articles to general tech overviews.

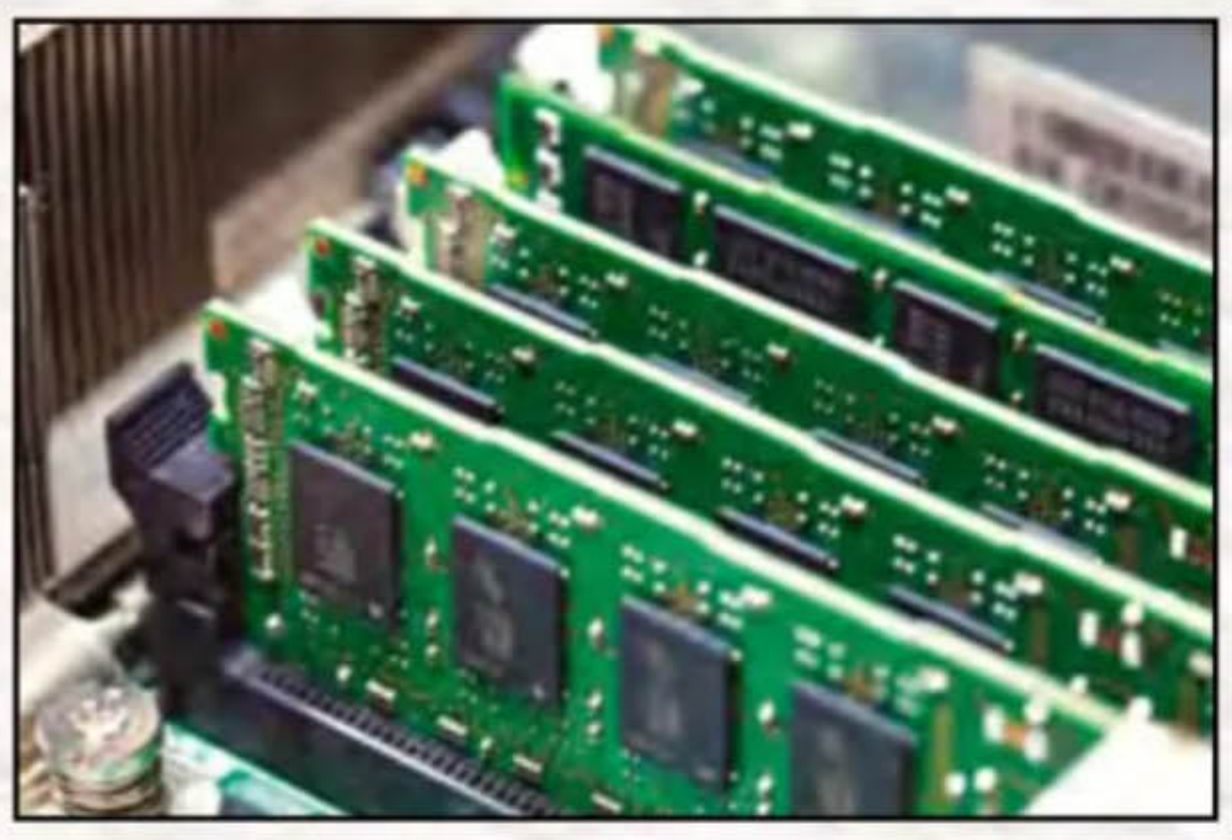
	Print	Combined	Online
6 months	$100 ≈£53 ≈€62	$110 ≈£59 ≈€68	$50 ≈£27 ≈€31
1 year	$195 ≈£105 ≈€120	$215 ≈£115 ≈€132	$95 ≈£51 ≈€58
2 years	$380 ≈£205 ≈€234	$415 ≈£223 ≈€255	$185 ≈£100 ≈€114

- *Combined subscriptions include both the printed magazine and online access.*
- *Prices are in Australian dollars (AUD) unless otherwise specified.*
- *Prices listed are for everywhere in the world except Australia and New Zealand. View* ***siliconchip.com.au/Shop/SubRates*** *for a full list of current prices.*
- *Price estimates are just that and should only be used as a guide, we primarily take payment in the form of Australian dollars.*

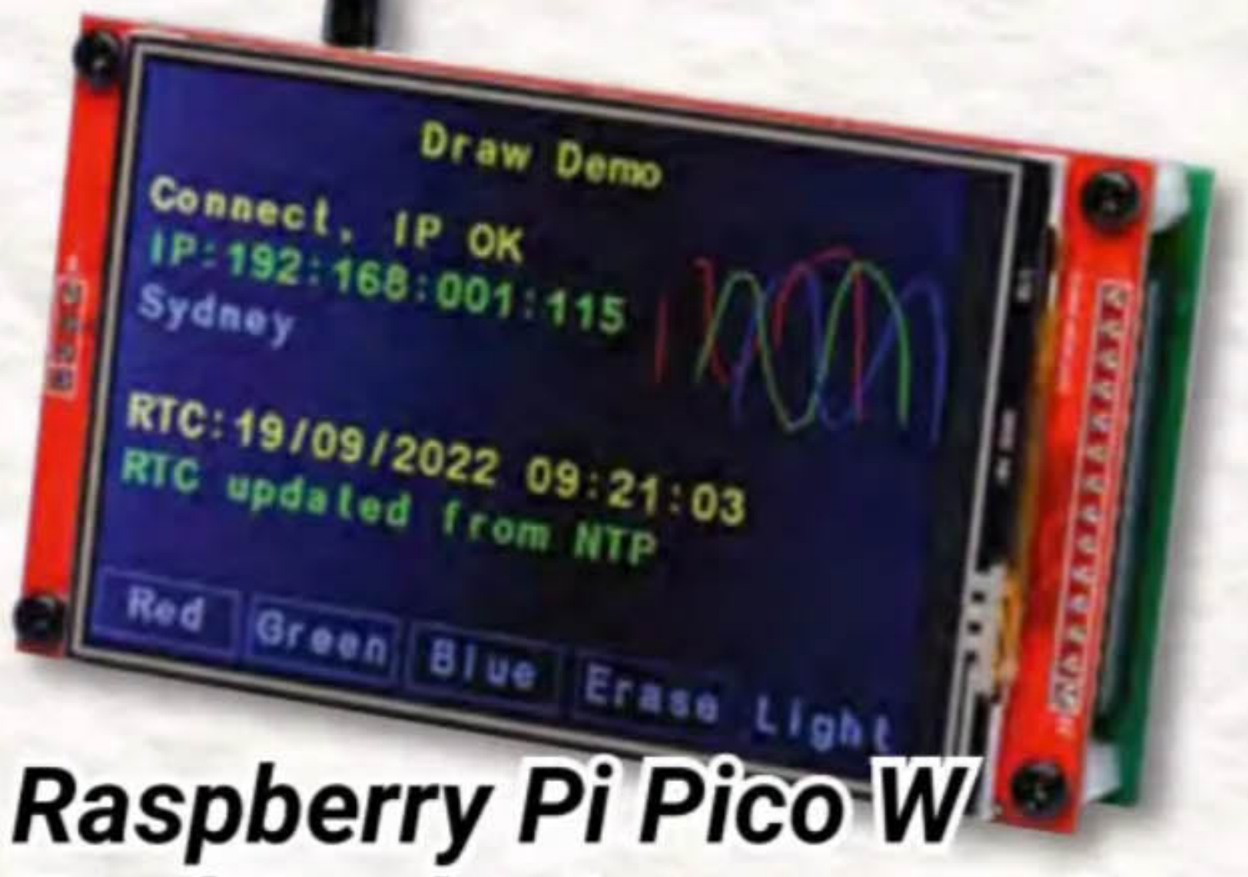
Try our Online Subscription – now with PDF downloads!

An online issue is perfect for those who don't want too much clutter around the house and is the same price worldwide. Issues can be viewed online, or downloaded as a PDF.

Published in Silicon Chip



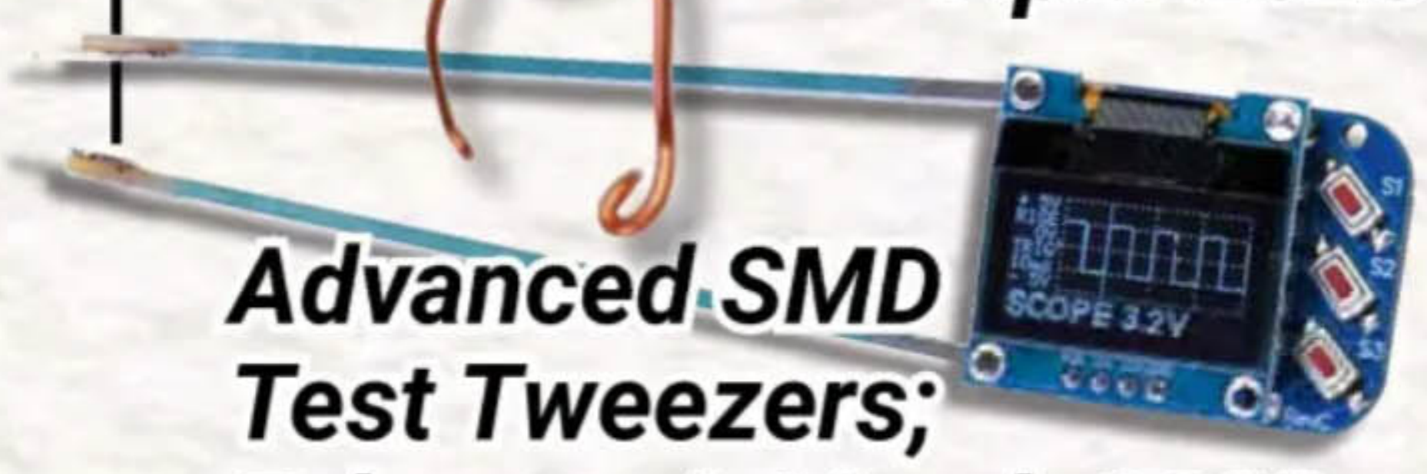
The History of Computer Memory; Jan-Feb 2023


Draw Demo
Connect, IP OK
IP:192:168:001:115
Sydney
RTC:19/09/2022 09:21:03
RTC updated from NTP
Red
Green
Blue
Erase
Light

Raspberry Pi Pico W BackPack; January 2023

Wideband Fuel Mixture Display; April to June 2023

Silicon Chirp; April 2023



Advanced SMD Test Tweezers; February & March 2023

To start your subscription go to siliconchip.com.au/Shop/Subscribe

DOWNLOAD YOUR PE/EPE BACK ISSUES

COMPLETE-YEAR collections – all 12 issues for each year!

From just £5.95 each

VOL 28 – January to December 1999 – **£5.95**
VOL 29 – January to December 2000 – **£6.95**
VOL 30 – January to December 2001 – **£7.95**
VOL 31 – January to December 2002 – **£7.95**
VOL 32 – January to December 2003 – **£7.95**
VOL 33 – January to December 2004 – **£7.95**
VOL 34 – January to December 2005 – **£8.95**
VOL 35 – January to December 2006 – **£8.95**
VOL 36 – January to December 2007 – **£8.95**
VOL 37 – January to December 2008 – **£8.95**
VOL 38 – January to December 2009 – **£10.95**
VOL 39 – January to December 2010 – **£11.95**
VOL 40 – January to December 2011 – **£12.95**
VOL 41 – January to December 2012 – **£14.95**
VOL 42 – January to December 2013 – **£15.95**
VOL 43 – January to December 2014 – **£16.95**
VOL 44 – January to December 2015 – **£17.95**
VOL 45 – January to December 2016 – **£19.95**
VOL 46 – January to December 2017 – **£22.95**
VOL 47 – January to December 2018 – **£27.95**
VOL 48 – January to December 2019 – **£27.95**
VOL 49 – January to December 2020 – **£29.95**
VOL 50 – January to December 2021 – **£32.95**
VOL 51 – January to December 2022 – **£34.95 – NEW!**

FIVE-YEAR collections – 60 issues!

From just £14.95 each

PE2832: Jan 99 to Dec 03 – £14.95
PE2933: Jan 00 to Dec 04 – £14.95
PE3034: Jan 01 to Dec 05 – £14.95
PE3135: Jan 02 to Dec 06 – £14.95
PE3236: Jan 03 to Dec 07 – £14.95
PE3337: Jan 04 to Dec 08 – £14.95
PE3438: Jan 05 to Dec 09 – £19.95
PE3539: Jan 06 to Dec 10 – £25.95
PE3640: Jan 07 to Dec 11 – £25.95
PE3741: Jan 08 to Dec 12 – £25.95
PE3842: Jan 09 to Dec 13 – £33.95
PE3943: Jan 10 to Dec 14 – £33.95
PE4044: Jan 11 to Dec 15 – £33.95
PE4145: Jan 12 to Dec 16 – £35.95
PE4246: Jan 13 to Dec 17 – £37.95
PE4347: Jan 14 to Dec 18 – £38.95
PE4448: Jan 15 to Dec 19 – £39.95
PE4549: Jan 16 to Dec 20 – £42.95
PE4650: Jan 17 to Dec 21 – £44.95 – NEW!

20 YEARS OF PE!

AMAZING BUNDLE OFFER
Every issue from 1999 – 2018

240 magazines for just

£59.95

PE2832: Jan 99 to Dec 03
PLUS!
PE3337: Jan 04 to Dec 08
PLUS!
PE3842: Jan 09 to Dec 13
PLUS!
PE3447: Jan 14 to Dec 18

Do you have the 15-year bundle?

Upgrade to the new 20-year bundle for just £14.95 with: PE15to20

Download your collections at: www.electronpublishing.com

Practical Electronics

Volume 53. No. 4
April 2024
ISSN 2632 573X

Editorial

Editorial offices
Practical Electronics
Electron Publishing Limited
1 Buckingham Road
Brighton
East Sussex BN1 3RA
Tel 01273 777619
Mob 07973 518682
Fax 01202 843233
Email pe@electronpublishing.com
Web www.electronpublishing.com

Advertisement offices
Practical Electronics Adverts
1 Buckingham Road
Brighton
East Sussex BN1 3RA
Tel 01273 777619
Mob 07973 518682
Email pe@electronpublishing.com

Editor Matt Pulzer
General Manager Louisa Pulzer
Digital subscriptions Stewart Kearn **Tel** 01202 880299
Online Editor Alan Winstanley
Web Systems Kris Thain
Publisher Matt Pulzer

Print subscriptions
Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom
Tel 01202 087631
Email pesubs@selectps.com

Technical enquiries
We regret technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine. We cannot provide data or answer queries on articles or projects that are more than five years old.

Questions about articles or projects should be sent to the editor by email: pe@electronpublishing.com

Projects and circuits
All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

A number of projects and circuits published in *Practical Electronics* employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

Component supplies
We do not supply electronic components or kits for building the projects featured, these can be supplied by advertisers. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

Advertisements
Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

Transmitters/bugs/telephone equipment
We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

Welcome to the future

As every electronics hobbyist will undoubtedly agree, nothing quite beats the thrill and satisfaction of powering up a newly-built project, and – fingers crossed – witnessing our latest endeavours bursting into operation for the very first time. In years gone by, this initial 'power-up' might have taken the form of a crackle or thump on a loudspeaker, the glow of vacuum tubes, a switch-on 'kick' on a moving-coil meter needle or the blink of a red LED showing a promising sign of life. Sometimes though, despite all the efforts we diligently invest in assembling our latest project, our toils might not go entirely to plan; instead of enjoying a triumphant power-up, we might instead see some 'magic smoke' escaping from an errant component or two, or, disappointingly, nothing at all happening. The principles of fault-finding and debugging will challenge our detective skills and are an integral part of our wonderful hobby, and it's immensely satisfying to overcome any teething problems and finally see our work springing into life as we'd initially anticipated.

I sometimes wonder what our predecessors would make of today's state of the art. Powering up our latest projects might see us greeted with a 'Hello world' exclamation on a liquid crystal display, or the boot-up menu of a crisp and clear OLED alphanumeric read-out, just before it proceeds to connect to the Internet. That's the marvellous thing about our hobby: there's always something new and exciting coming along to pique our interest and challenge our constructional skills. Our workbench is a place where we can escape to and keep our brain busy for a time, while an increasingly noisy and uncertain world runs its course outside.

Here at *Practical Electronics* we do try to move with the times by bringing you modern projects that will supply the enjoyment and satisfaction that we know our readers have a thirst for. We try to cater for all levels, though, including our excellent educational features such as *Circuit Surgery* and *Teach-In*, instructional material of the highest quality that you won't find anywhere else. One thing's for sure, our forebears would blink in total disbelief at what we can now achieve using just a handful of miniscule parts and some very clever coding. Over to you, readers – welcome to this month's *Practical Electronics*.

Matt Pulzer
Publisher

Wait! What? Really?

Techno Talk

Max the Magnificent

I first read *Ready Player One* deep in the mists of time we used to call 2011 (which is now 13 years ago – give me strength!). At that time, the technology depicted in this tale was the stuff of science fiction, but we are getting closer day-by-day.

When the combination of artificial intelligence (AI) and mixed reality (MR) comes to fruition, it will change the way we interface with our systems, the world, and each other.

Keep it real!

As John Lennon famously noted, 'Reality leaves a lot to the imagination.' In fact, there are multiple flavors of reality. At one end of the reality spectrum we have the real world, which we might call 'physical reality' (PR). At the other end we have virtual reality (VR) in which everything we see is computer generated.

Most people these days are familiar with the term augmented reality (AR). Many believe this to involve only a visual view of the real world superimposed with text or graphics. More generally, and quoting myself, we might say that AR refers to 'an interactive experience in which objects that reside in the real world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory [relating to a sensation such as pressure or warmth], and olfactory.'

The conceptual opposite to AR is diminished reality (DR), which involves diminishing or removing information or stimuli from the real world. One example would be if you were at a cocktail party and your DR-enabled system faded down all the voices except that of the person with whom you are conversing. Another would be to change most of your view of the world into grayscale, leaving only the portion of interest in colour. Think of a *Where's Waldo* image. It would be a lot easier to spot the little rascal if he was the only portion of the picture in colour. Similarly, if you were looking for your friends in an environment containing thousands of other people – like an open-air concert, for example – it would be advantageous if your AI+AR+DR headset presented everything but your friends in black-and-white.

And then there's augmented virtuality (AV) in which a predominantly virtual environment is augmented with real-world objects or people. Because all this is a bit of a mouthful, we use the term mixed reality (MR) to embrace AR, DR, VR and AV. Don't get me started on hyper reality (HR), but feel free to search for 'Hyper Reality' on YouTube.

More cheese, Gromit?

I was first exposed to VR in the form of military systems 20+ years ago. At that time, I thought a consumer-level experience of any significance was years away in my future. So, you can only imagine my surprise and delight when the Oculus Rift VR headset launched in 2016. Suffice it to say I was at the front of the queue for one of those bodacious beauties.

Of course, being tethered to a large computer and having to use external sensors was a bit of a downer, which is why I traded my Oculus Rift for an Oculus Quest (now Meta Quest) VR headset as soon as it came out in 2019. More recently, I splashed the cash for a Meta Quest 3 MR headset, which was released in October 2023. This little scamp is sitting on my desk singing its siren song as I pen these words.

There are myriad mixed reality applications available for the Quest 3. One of my favorites involves Wallace and Gromit. Assume you are standing in the middle of your family room. The headset has forward-looking colour cameras whose video streams are presented on the displays that reside in front of your eyes. As a result, you see the same thing you would if you took the headset off… apart from the virtual electric toaster in front of you and the virtual jam-dispensing, steampunk-looking gun in your hand. When a slice of toast is ejected from your toaster, you attempt to hit it with a blob of jam fired from your gun. If you miss, the jam carries on its trajectory until it hits (and sticks to) the ceiling or wall. *Brilliant!*

The noblest of fruits

The current state-of-the-art in consumer MR is the recently introduced Apple Vision Pro. (It's tempting to speculate that Henry David Thoreau was channeling the future when he proclaimed, 'Surely the apple is the noblest of fruits.')

When people are first exposed to Vision Pro publicity shots, they often assume that this device features a transparent visor because it looks like you can see the wearer's eyes. In reality (no pun intended), the visor is opaque. This feature – which is called EyeSight – starts with a 3D facial scan of the person wearing the headset. In addition to other tasks, four eye-tracking cameras inside the headset are used to control where the virtual eyes are looking on the outward-facing display.

I would *love* to own a Vision Pro, but it's way outside my price range. A Quest 3 headset costs around £400 in the UK. By comparison, a Vision Pro will lighten your wallet by a hefty £3,500 give-or-take (Eeek!).

Wait! What? Really?

It's probably fair to say that most people might be a tad embarrassed to be out and about – perambulating through town or strolling round a supermarket, for example – sporting a Quest 3 or Vision Pro style headset. Having said this, no one would care if everyone else was doing the same thing. Hopefully, it won't be long before we have display technologies like those being developed by Kura (**www.kura.tech**), which look like regular glasses but sport 50,000,000 MR pixels per eye.

I have just seen something else that has me quivering in excitement. This comes to us from Zinn Labs (**www.zinnlabs.com**). It's a set of glasses frames (no lenses) equipped with a forward-looking camera that sees what you see and eye trackers that can determine what you are looking at. If you visit Zinn's website, you can see these glasses connected wirelessly to ChatGPT. All you do is look at something, ask a question about it, and let the ChatGPT-generated answer tickle your lugholes.

I've said it before, and I'll say it again – the future is closer than we think!

Win a Microchip Explorer 8 Development Kit

Practical Electronics is offering its readers the chance to win a Microchip Explorer 8 Development Kit (DM160228) – and even if you don't win, receive a 15%-off voucher, plus free shipping for one of these products.

The Explorer 8 Development Kit is a full-featured development board and platform for 8-bit PIC microcontrollers (MCUs). This kit is a versatile development solution, featuring several options for external sensors, off-board communication and human interface. Additionally, it offers ample room for expansion, making it an excellent solution for developers and engineers looking for a tool with the largest number of supported 8-bit PIC MCUs.

The Explorer 8 Development Kit is the latest offering in a long line of enterprise-class tools for 8-bit PIC MCUs. The board is an evolution of the popular PIC18 Explorer Board, and has been updated to take advantage of our most modern features:

- On-chip Core Independent Peripherals
- Compatibility with all PIC MCUs
- 8-, 14-, 20-, 28-, 40/44-, 64- and 80-pin footprints

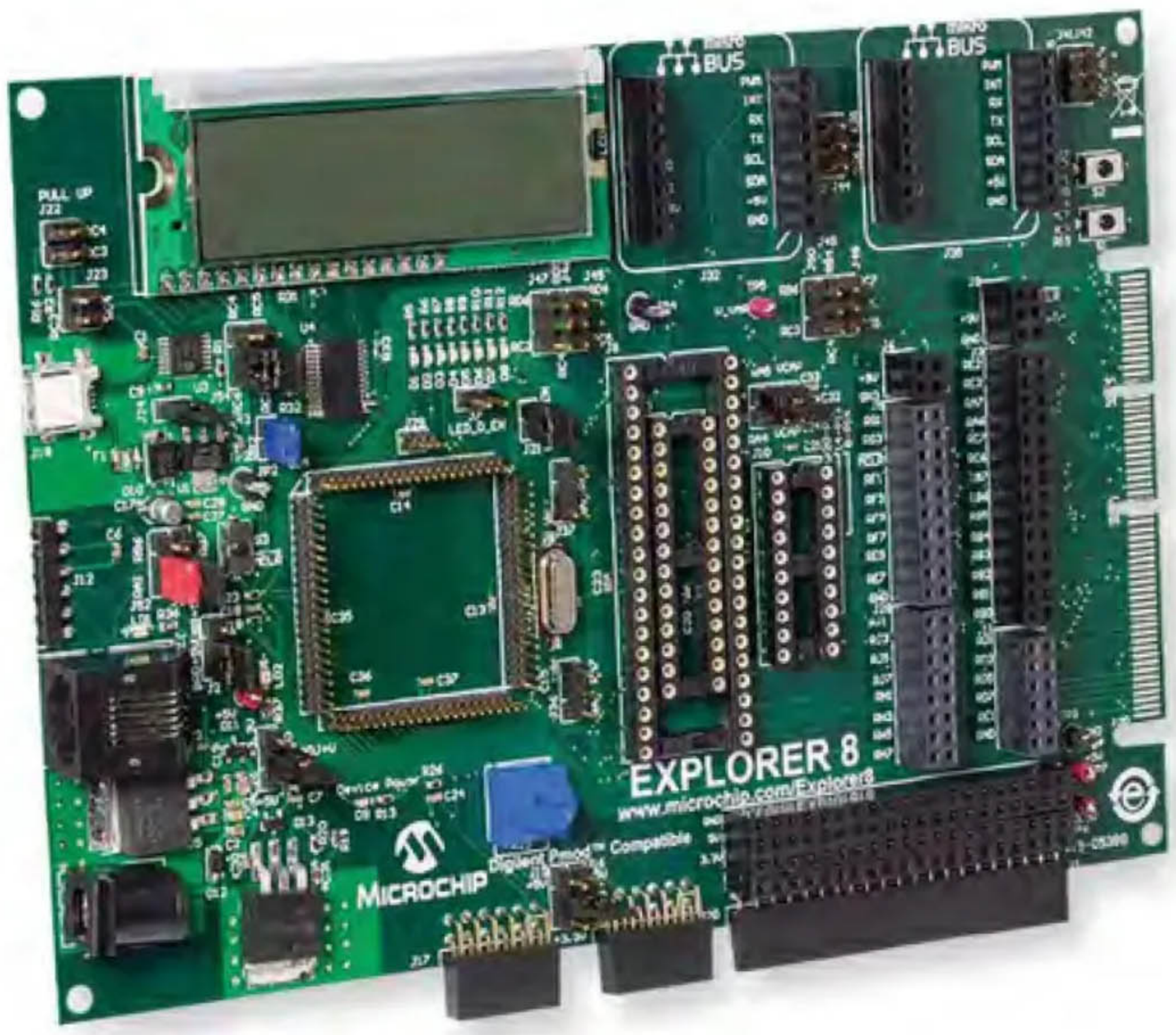
Worth $123.99 (approx £97.50) each

With the Explorer 8 Development Kit, you can have a single development platform that allows you to update a legacy design while supporting our newest MCU offerings.

Microchip designed the Explorer 8 Development Kit to be your main tool for 8-bit development with support for a wide variety of functions. On-board components enable easy development of human interface, power conversion, Internet of Things (IoT), battery charging and other applications using a powerful 8-bit PIC MCU.

Additionally, the Explorer 8 Development Kit also has a large capacity for expansion. Two Diligent Pmod interfaces, two MikroElectronika Click board sockets and two custom expansion headers allow you to quickly adapt as your development needs change. The Explorer 8 Development Kit supports several program and debug solutions, including the PICkit 5, ICD 5 and MPLAB REAL ICE In-Circuit Emulator.

Free-to-enter competition



Microchip
Win a Microchip Explorer 8 Development Kit

Microchip
DM160228

How to enter

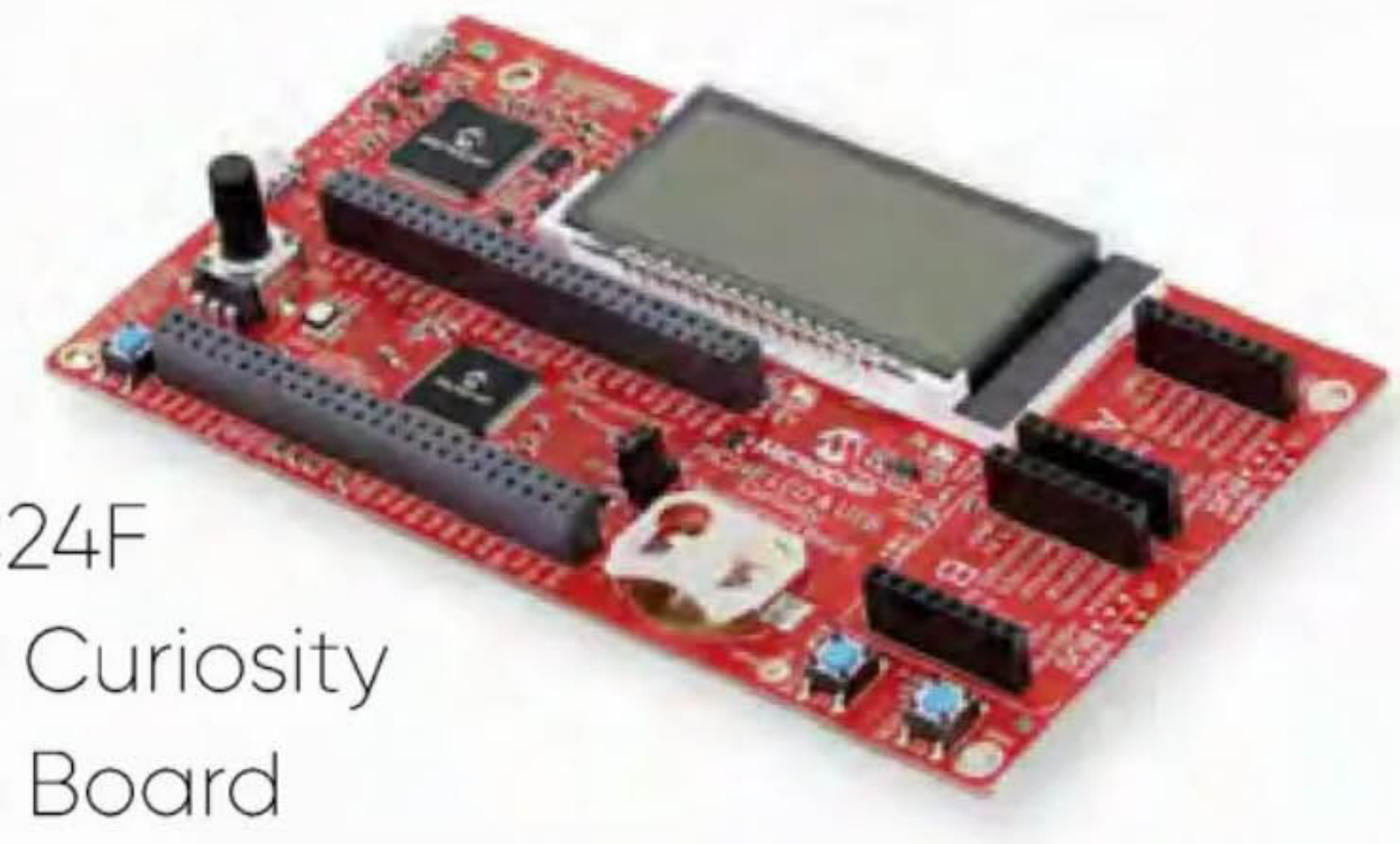
For your chance to win a Microchip Explorer 8 Development Kit or receive a 15%-off voucher, including free shipping, enter your details in the online entry form at:
https://page.microchip.com/PE-Exp8.html

Closing date

The closing date for this offer is 31 March 2024

December 2023 winner

Barry Lewis



Barry won a Microchip PIC24F LCD and USB Curiosity Development Board

Net Work

Alan Winstanley

Our column covering Internet-related trends, and more besides, bags a label printer from China at a bargain price. Plus some current and new smart home products by TP-Link's Tapo are examined. There's also news on the SMR power generation front.

It's now second nature for electronics hobbyists to source materials, tools, components or printed circuit boards directly from China, and websites such as AliExpress, Banggood and Temu are falling over themselves to attract our business. These websites sell a myriad of products, and they push heavy discounts or 'new user benefits' in order to hook new customers. It's no secret that these Chinese super-sites are slugging it out for trade, and after several months of being bombarded by their pop-up adverts, I finally buckled and decided to try the newcomer Temu website for the first time.

Temu's online store is very slick and quite overpowering, but what caught my eye was an introductory 'first time customer' offer on a thermal label maker, specifically the Phomemo-brand M110 handheld Bluetooth printer. I've had my eye on such a machine for quite a while, because they are handy self-contained rechargeable devices that can print, among other things, barcodes, QR codes and labels for identifying cables. They will also print die-cut or continuous labels on small rolls. Although the makers call it a 'business' printer, it's the sort of thing that can soon find uses in the home and workshop, and various coloured and transparent labels, either diecut or continuous, are sold too. Also available is the model M220 which has a wider print head and larger battery.

The M110 by Phomemo is a self-contained Bluetooth and USB thermal labeller that prints on both polymer diecut and paper labels.

These labellers have a street price of about £45 ($55) but Temu was offering them for just £11 ($13), and four extra rolls of labels cost about the same. Thus, their thermal labeller with refills would come in at about £25 ($30) delivered which, unlike AliExpress, included sales taxes with no hidden extras. Was this too good to be true? There was only one way to find out, and a smooth PayPal transaction was quickly completed in a few mouse-clicks.

Testing Temu

Six days later my first Temu order arrived in England all present and correct, in the usual kind of barcoded polybag. The rounded design of the M110 printer takes up little space, and mine was moulded in blue ABS sporting a large logo 'window' that looks like some sort of display screen (it isn't). There's a two-colour 20mm OLED status display, which is small but crystal-clear in operation. The printer contains a 1200mAh lithium battery which is charged through a USB-C port and, as per the current trend, you provide your own 5V mains charger.

Labels are designed with the free Print Master app for iOS or Android. Pairing the Bluetooth printer was trouble free, and after loading the supplied roll of labels I was printing labels within a minute or two. The resolution quality was very good, and I was surprised by the comprehensive labelling software options provided: apart from plain text, it will print QR codes, EAN and UPC barcodes, as well as tables, monochrome images and icons. In fact, it goes even further by offering OCR text capturing and voice recognition, which worked very well. Label content can be dragged around or resized, and a print history is stored to allow future reprints.

Initially, I found that designing labels was better done on a tablet than a smartphone. Either way, calling up (tap, swipe, double tap … etc) the text editor window to make changes was hit and miss, frustratingly so at times, and the text input field had a low contrast and occupied a tiny area on my phone and tablet screens. The software was a bit counterintuitive, but I managed to muddle through.

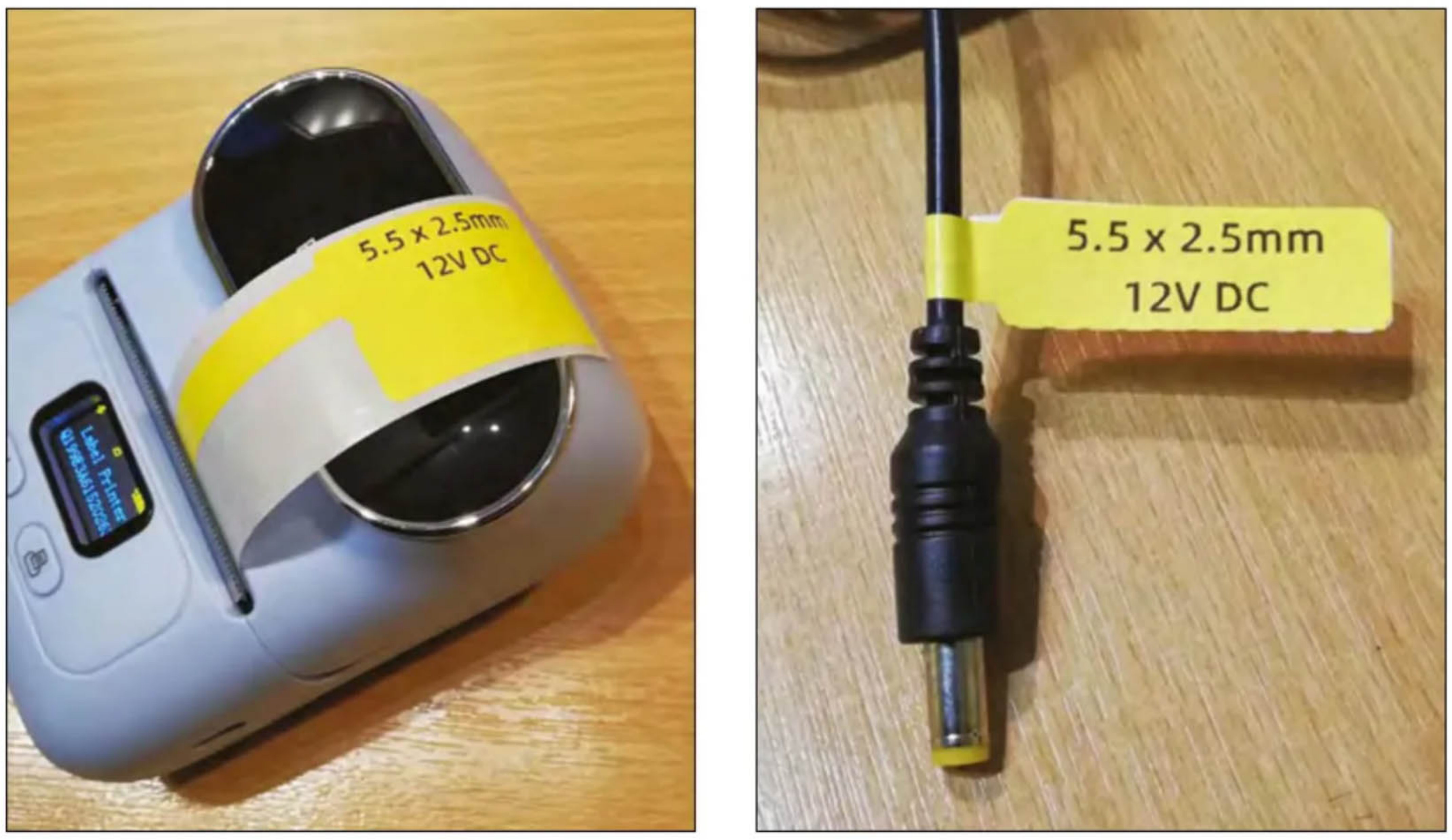
Printing QR codes using the M110 labeller.

This way of working will be fine for printing labels on the go or in awkward locations, but I eventually realised that the makers had made a strange omission: although the spec. sheet states Bluetooth, Android and iOS compatibility, not mentioned anywhere was the fact that, as I later discovered, printer drivers for 32-bit and 64-bit Windows and macOS are downloadable from the maker's website, along with free label creation software. There is also a Linux driver. Having installed the latest Windows driver and free 'Labelife' software on my PC I was soon downloading labels to the printer via a USB-C cable. The printer firmware can also be updated that way. Using the desktop software, the required label size can be chosen from the Templates menu and then text, images, QR codes and so on can be added. Designs can be saved to disk for future use.

Cable label capable

Confusingly, the makers describe direct thermal labels as 'paper' but the 40 x 30mm diecut labels supplied are made of polymer film, which is resistant to tearing, abrasion, chemicals and water. They are good quality labels with semi-permanent adhesive, and they can be peeled off again without leaving a residue. The adhesive is not freezer-proof though (as my frozen pies will testify).

Cable identification labels made from the same plastic material are available



Cable identification labels can also be produced on special diecut blanks.

in various colours and are handy for electricians, network engineers and constructors. These have a printable area of 35mm x 12mm on each side, and the wraparound is good for cables up to 10mm diameter or so (12mm max), and although they are tear-resistant they are not indestructible. It was easy to print labels in the app, and the narrow section is then stuck around the cable before the main printed area is folded over to stick it all together (see photos). This idea works very well. Although the Windows software has a template for them, strangely the printable area of the cable label is not defined, and I wasted quite a few labels due to misalignment. The mobile Print Master app is a better choice here.

A number of other printers is sold by Phomemo, including a fun sticker printer and a 6 x 4 shipping label machine. Other labels in the range include transparent and decorative ones for crafts or hobbies, so it's worth browsing around the maker's website for ideas. You can buy labels direct using PayPal from **https://phomemo.com** and watch for a pop-up discount voucher code when you enter the site. Printers and blank labels also appear on Temu, Amazon and other sites.

Tapo dancing

Regular readers will know that I've been trying the Tapo range of smart home controls for quite some time. Tapo is produced by TP-Link and it gives us an indication of emerging smart home trends. This interesting and ever-expanding brand is aimed at domestic users looking to add affordable smart controls to their home. Several Tapo bulbs and smart plugs (mains sockets) are in regular use by the writer which work well enough most of the time, although sometimes they can be temperamental, and I have to reboot them to restore functionality. As Tapo is intended to control small domestic appliances, personally I would avoid using them with, say, electrical heating or motorised machines like an air conditioner or hefty white goods like a washing machine, or garage door motors, alarm systems or similar more demanding applications.

Some Tapo smart plugs include an energy monitoring feature, with the corresponding runtime and usage costs displayed on the Tapo app. They double as compact, handy programmable time switches as well, and they can time up or down. In 'Away' mode they can act as a burglar deterrent by switching, for example, lighting randomly during predetermined periods. After installing the TP-Link 'skill' on Amazon's Alexa app (Google Home is also supported), I now have voice control of lighting and TV power too, which works sufficiently well 95% of the time. It's also possible to control them remotely with your app, and share the control of certain Tapo devices with up to eight friends and family.

Coming soon is an updated Tapo smart socket, the P110M which includes an energy-monitoring chip that it claims provides overload protection. This smart device will be Matter compatible (see *Net Work*, April 2023). Matter is the slowly evolving new LAN protocol that promises to make the installation of smart devices as simple as 'plug and play'. 'Simply

matter
tapo
MAX 13A

The Tapo P110M is a smart plug with energy monitoring that is Matter compatible. It's rated 13A for the UK.

HAMMOND®
1551W IP68 miniature enclosures
Learn more: hammondmfg.com/1551w
uksales@hammondmfg.com
01256 812812

scan the provided QR code with your smartphone and the Matter device will be automatically configured to your home network. Enjoy your smart home products with ease right out of the box!' as TP-Link puts it.

It does Matter

Matter aims to enable compatible Matter smart devices talk to each other, even if the home itself goes offline (eg, due to a broadband outage). A host of Matter-compatible devices are scheduled for early release by TP-Link, and anything that simplifies an often technical and onerous set-up routine can't come soon enough. For more details visit: **https://bit.ly/pe-apr24-tpl**

TP-Link's 'Kasa' range of smart devices predates Tapo and seems more robust. Until recently, Tapo and Kasa required their own apps and were not interoperable, but following an update the Tapo app can now control both device ranges. To test this out, I tried adding an old TP-Link Kasa smart socket (HS100) to my Tapo app – this was previously impossible to do, and although the setup routine was a throwback to the earlier days of using smart home devices, it did install successfully and proceeded to update its firmware. A more compact smart socket (the Kasa KP105) is due out soon, while the new KP115 smart plug includes energy monitoring. Both devices have a current rating of 13A (UK use). Not available in the UK yet, the Kasa KP125M is a new compact size Matter-compatible smart plug.

In Britain, the Kasa brand also includes smart radiator thermostatic valves to control central heating, and there is a whole raft of security cameras and video doorbells to choose from in TP-Link's rapidly expanding repertoire of home-networked devices. TP-Link does seem to be investing in this product line and it's one worth keeping an eye on to follow trends and developments.

The H100 Smart Hub with Chime works on 868MHz with compatible smart sensors and peripherals. It continues working locally even if the home broadband is down.

A hub of activity

Also part of the Tapo range is the 'Smart Hub' (H100) which forms the heart of a sub-group of Tapo smart sensors, switches and buttons. It's a plug-in mains-powered hub and its main feature is its built-in audio chime and ability (in theory) to connect to up to 64 devices. The spec. sheet highlights that it uses 2.4GHz Wi-Fi, but it also uses the less crowded 868MHz frequency (for UK models) to link with compatible Tapo smart devices, which TP-Link claims improves battery life ten-fold. (The same frequency band has proven extremely reliable when used by my Ecowitt weather station sensors, described in previous issues.) They are 'cloud free' in the sense that they will continue operating locally, even if the Internet connection goes down, provided the mains supply is still running. The app also lets you share functionality with designated friends.

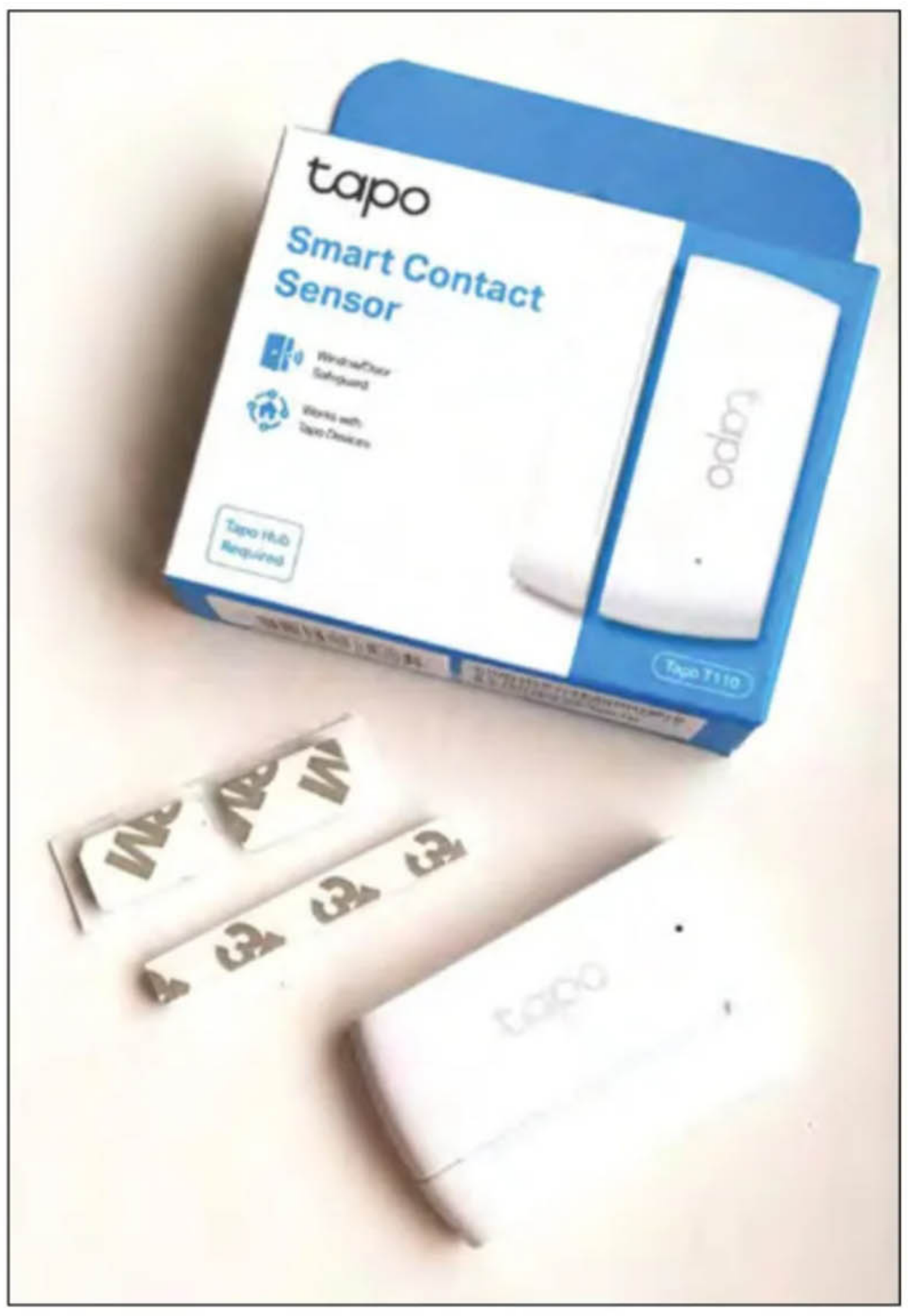
I tried the H100 hub with a Tapo Smart Contact Sensor T110 by adding them as devices on the Tapo app, then firmware updates were downloaded automatically. Setup was not as slick as I'd hoped for, and it took several attempts to configure everything successfully (I gave up at one point). As a surprise bonus, ten minutes later my Amazon Echo Show notified me that it detected a new device (the contact sensor) being added on the TP-Link 'skill'.

The Tapo app lets you choose from 19 hub chimes such as doorbell 'ding-dong' or alarm tones which are clear and strident (four volumes, up to 90dB). You then link compatible sensors or switches to the Tapo hub. So-called 'Smart Actions' then enable alarms or chimes to sound when certain triggers are detected. In my case, an alarm will sound when the contact sensor is opened or closed (my choice) and a delayed action from seconds to hours can also be set. Its hours of operation can also be dialled in like a time switch, and all triggering actions are logged in the app.

I had to invest quite some time in the app before everything would function as desired: there was no quick 'plug and play' here. I eventually discovered that configured Smart Actions were silently stored in the app under 'Automation' where they could be toggled on or off or deleted. This probably explains the inconsistencies in operation that I experienced.

Sticky sensors

The contact sensor itself is a two-part set containing a magnet and it's not particularly sturdy. The only mounting option is via the pair of sticky pads supplied, which seems a missed opportunity; you can't screw-mount them onto a wooden door frame, for example. Also available are a PIR motion detector, a temperature/humidity sensor with LCD screen and a smart water leak sensor.



The T110 magnetic sensor uses the H100 chime to warn of open doors or windows. It only has sticky pad fixings.

The Tapo Smart Button (S200B) allows human-interactive control of other Tapo smart devices such as light switches, bulbs with a single or double tap, and a rotating action accesses more functions (eg, a dimmer). It could also be used as a nurse-call alarm for elderly care, or it could set, say, 'Away' mode when you leave the building.

A 3D-printed doorbell adaptor plate for the S200 button is sold on eBay (stevedark001).

Keen to sample the performance of this budget-priced range, I've now added a Tapo smart rotary button (S200B) and a smart water leak sensor (T300) onto my home network for trials. Iin next month's *Net Work* I'll report back on how well these low-cost smart devices are faring in practice.

A wallplate version (S200D) would replace a wall light switch, but I'm not sure how robust it would be, either mechanically or network-wise. It's also suggested for sheltered doorbell use, the hub sounding the doorchime. (Note the button would be easily pocketed.) An eBay user (stevedark001) sells a 3D-printed base and holder for Tapo smart buttons that resembles a bell pushbutton: Steve tells me it came about because visitors didn't realise his Tapo button was in fact the doorbell!

For more demanding users, the Smart Hub (H200) hub uses a mains adaptor instead and has an Ethernet port and microSD storage (512GB max), suitable for up to 64 devices and up to four compatible Tapo security cameras or video doorbells. Chimes and sensor detection are also included. More details and an online shop are at: **www.tapo.com**

Hinkley Point C – Helping Britain Achieve Net Zero

Putting the lid on: the world's largest crane, Big Carl, placed a 245 tonne lid onto the Hinkley Point C Unit 1 nuclear reactor last December. A second reactor will follow.

Power to the people

Work continues to make the UK's nuclear power generating industry fit for the 21st century. At the site of the Hinkley Point 'C' nuclear power station in Somerset, southern England, the world's largest crane, the 750 tonne 'Big Carl', is doing the heavy lifting and last December it topped off the Unit 1 steel containment building with a 47-metre diameter, 245-tonne lid. The French-built reactor will be installed this year and is the first new (UK) reactor to be built in over 30 years. A second reactor is also under construction which, thanks to the experience gained in rebuilding an entire industry, should take 20-30% less time to complete, says EDF Energy. The project has been plagued by delays, Covid and cost increases ever since it was approved in 2016, and presently the £26bn ($31bn) project is due to come on stream in 2027. (Attention will then turn to building another new nuclear station, Sizewell C.) An honest appraisal of progress is given by EDF in the video at: **https://youtu.be/rxhHTpnKX8Y**

As regular readers will recall, I've previously described the concept of the Small Modular Reactor (SMR), which offers the power generation sector a faster, cheaper route to bringing nuclear power stations on stream. The idea of SMRs is that key elements would be manufactured off site, before being delivered by road for final assembly at smaller power stations. SMRs could also be located in more isolated or rural areas which will improve distribution. This factory-built, production line approach would improve efficiency and help keep the lid on costs. The International

Terrington Components

- Project boxes designed and manufactured in the UK.
- Many of our enclosures used on former Maplin projects.
- Unique designs and sizes, including square, long and deep variations of our screwed lid enclosures.
- Sub-miniature sizes down to 23mm x 16mm, ideal for IoT devices.

MADE IN BRITAIN

www.terrington-components.co.uk | sales@terrington-components.co.uk | Tel: 01553 636999

Westinghouse has been chosen to supply four SMRs for a privately funded British nuclear power station in Teesside.

Atomic Energy Authority states that there are more than 80 SMR designs under different stages of development in 18 Member States, including the *Akademik Lomonosov* floating power unit moored in the Russian Federation, which started generating in 2020 (as shown in *Net Work*, March 2022).

Britain's Rolls-Royce is one of several blue-chip manufacturers scrambling to be at the forefront of SMR development, and last October the company was keen to remind us that it 'had the only SMR technology in a European regulatory approval process, putting us almost two years ahead of any of our competitors. Securing a domestic contract is vitally important to unlock the enormous global export potential of our clean energy technology.' The company is also keen to 'reinforce its position as the SMR design that's furthest advanced towards being deployed in the UK.'

Scoping for sites

Rolls-Royce SMR is known to be scoping potential factory sites for manufacturing modules, including one in the author's locality, but its SMR plans have yet to be granted full approval. Rivals in the SMR race include America's NuScale, who were the first to have their proposals signed-off ready for production, as I reported in previous columns. NuScale has majored on the claim that they are ready to go with an approved design, and the firm says it has now started production of forgings for the first modules.

Despite this, last November NuScale, by mutual agreement, pulled out of a major contract for its SMRs that were destined for a project in US state Utah, citing rising costs and non-viability. A statement by Dr Edwin Lyman, Director of Nuclear Power Safety at the non-profit Union of Concerned Scientists is more telling: he cites 'regulatory corner-cutting' and claims that 'NuScale made several ill-advised design choices in an attempt to control the cost of its reactor, but which raised numerous safety concerns.' He goes on to say how it 'reflects the fragility of the advanced nuclear power industry in the US, which has been driven by an oversupply of reactor developers and a lack of genuine demand'. The full statement is at: **https://bit.ly/pe-apr24-ucsusa**

Despite this major setback, NuScale remains very upbeat and positive about the future, citing future projects including ones overseas that are in the pipeline.

The scramble to get SMRs to market continues, and back in Britain a major step forward has been taken by a privately financed SMR project to build four SMRs in Teesside, in northeast England. Actual power generation using SMRs is still some ten years away and the Teesside plant is expected to start producing in the early 2030s. The supplier of the four reactors is neither Rolls-Royce nor NuScale, though, but America's Westinghouse. Rolls-Royce has however been chosen as the preferred supplier for a site in West Cumbria and has its sights on delivering SMRs to the Netherlands as well.

Rolls-Royce SMRs could also eventually be dedicated to producing clean fuels such as hydrogen. With the market for electric vehicles faltering, it's worth noting that Honda and GM recently signed a joint venture to produce hydrogen fuel cells, which may ultimately find their way into vehicles of the future. Meantime a pilot project to heat an entire town, Redcar in northern England, using hydrogen has been shelved because the main source of hydrogen supply will not be available. Trials at Whitby and Ellesmere Port had also been shelved earlier.

At the other end of the energy scale, for my own part, living in a rural area I still sometimes struggle with occasional power cuts. I'd just finished writing last month's column when I had yet another unscheduled outage. Happily, my newly installed 9V and 12V uninterruptible power supplies (see previous months) held up magnificently, keeping my PC, router, switch and mesh hub running for over half an hour. It's enough to make anybody want to live off-grid!

Battle of the bots

The industry obsession with AI chatbots continues as Microsoft integrates Copilot into the Edge browser (described last month) and, eventually, the Windows desktop. Google's experimental Bard chatbot was touted as 'an early AI experiment that can boost your productivity, accelerate your ideas and fuel your curiosity' but was recently rebranded as *Gemini*, so you will soon be able to 'chat with Gemini' when searching online. A Google search reveals that the astrological sign Gemini is named after the constellation twins Castor and Pollux, so now you know!

Bringing your pet dog to work was an idea that Amazon embraced at its Seattle HQ in its earliest days. Some 7,000 pooches are now registered across 40 office blocks, and more than 500 dogs are in the offices at any one time, Amazon says. It's why you see a picture of a dog on an Amazon 'Uh oh' error page. Amazon is developing its own AI bot called *Rufus*, named after the very first dog (a Welsh corgi) ever to grace the office, where he was clearly a well-loved member of the Amazon team: **https://bit.ly/pe-apr24-amz**

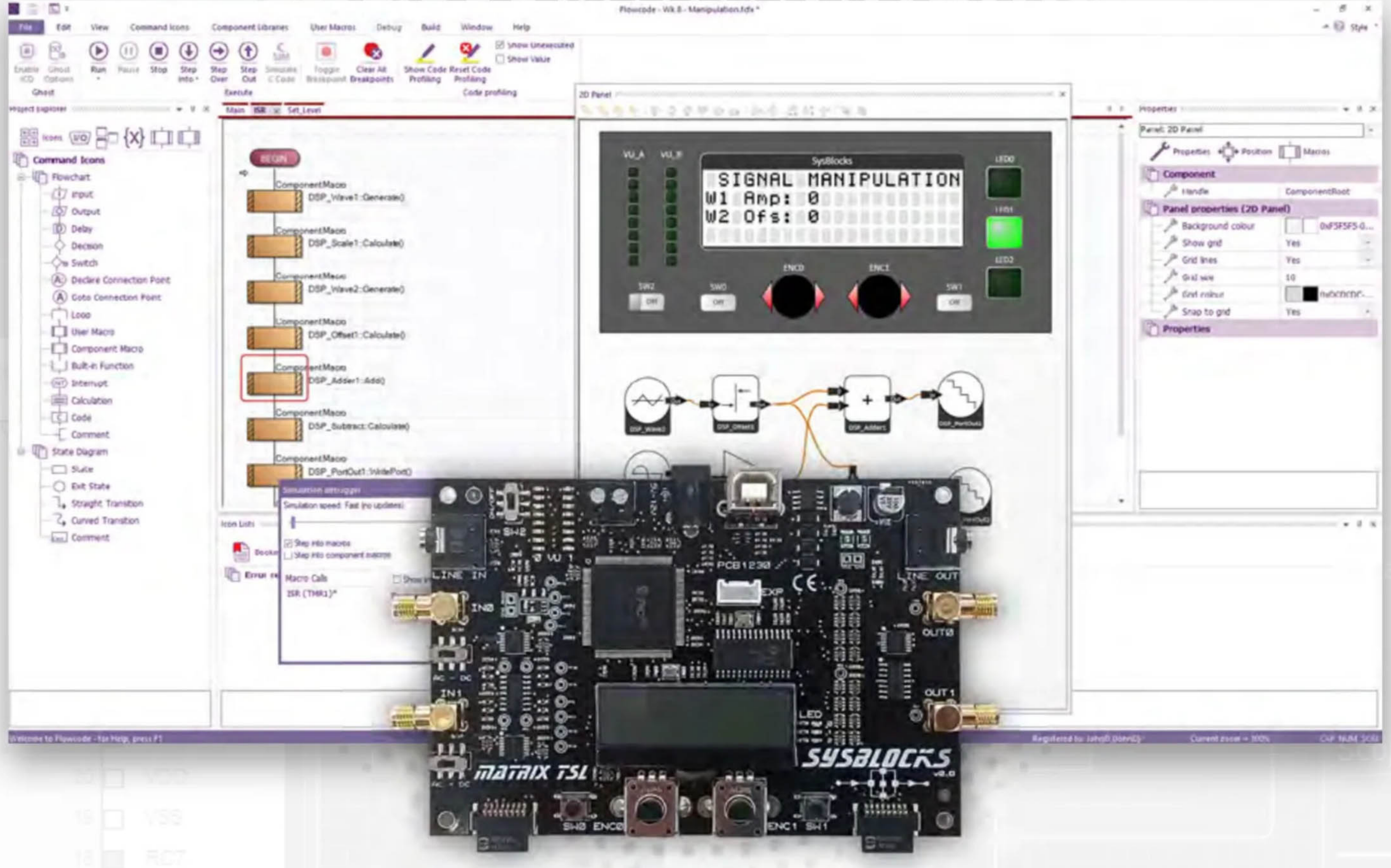
Amazon's Rufus AI is a shopping assistant 'bot that is currently undergoing beta house-training with select mobile app users in the US. He'll be let off the leash across the US later this year, and presumably Rufus will obediently roll over into the UK in due course. (That's enough dog puns.)

Mozilla, the name behind the Firefox browser and Thunderbird email client, has launched a hobby project sharing journal called 'DidThis'. It's aimed at crafters, hobbyists, constructors, home chefs, writers, knitters, woodworkers, wedding planners, car restorers... anyone who's got a personal project on the go and would like to share its progress and everything they experienced along the way. It could be a brilliant way of sharing electronics projects without having to resort to the vagaries of Facebook or a blog – you just use a smartphone to upload images. At the moment, DidThis works on iOS only, but an Android version is promised. Stories are kept 'private' by default. I think it's a terrific idea – more details are at: **Didthis.app**

That's all this month – do remember that all the above web links are ready-made for you on the *Net Work* blog at: **www.electronpublishing.com**

The author can be reached at:
alan@epemag.net

Sysblocks



SIGNAL MANIPULATION
W1 Amp: 0
W2 Ofs: 0
SysBlocks
BEGIN
DSP_Wave1::Generate()
DSP_Scale1::Calculate()
DSP_Wave2::Generate()
DSP_Offset1::Calculate()
DSP_Adder1::Add()
DSP_Subtract::Calculate()
DSP_PortOut1::WritePort()
MATRIX TSL
SYSBLOCKS
LINE IN
LINE OUT

CREATE HIGHLY FUNCTIONAL ELECTRONIC SYSTEMS USING PIC32 TECHNOLOGY TO MANIPULATE ANALOGUE SIGNALS USING FLOWCODE 10.1 AND SYSBLOCKS. FANTASTIC FOR DIGITAL AUDIO APPLICATIONS.

Flowcode Pro, PIC32 pack with Sysblocks board, cables and accessories bundle £610 + VAT
Flowcode with PIC32 pack with Sysblocks board, cables and accessories bundle £330 + VAT

10% off for PE readers using code: **PESYS10**

FLOWCODE 10.1 WITH MORE THAN 20 MICROCONTROLLER TARGETS INCLUDING ARDUINO UNO, PIC18877, RPI2B, AND ESP32 IS NOW FREE FOR HOBBYISTS.

☎ +44 (0) 1422 252 380 — ✉ sales@matrixtsl.com

www.flowcode.co.uk

Basic RF Signal Generator

AD9834 DDS SIGNAL GENERATOR
ON
FREQ

My AM/FM/DDS Signal Generator design (May 2023) is a very convenient piece of test equipment, but it's overkill for many tasks. If you only need a basic test signal from 10Hz to 25MHz, then this Generator is just right – it's compact, cheap to build and doesn't involve many parts.

By Charles Kosina

This design came about because my *Q Meter* (*PE*, January 2024) needs a 100kHz to 25MHz signal at close to 0dBm to function. Many constructors may already have a suitable signal generator, such as my design from *PE*, May 2023. Still, I decided to create a simpler version that does the job with minimal components and at a lower cost.

A DDS design is the most sensible option, and the Analog Devices AD9834 is a good DDS chip, but it costs £15 plus delivery. It makes far more sense to purchase a ready-made module, which costs less and comes with most of the necessary parts already assembled onto a PCB.

There are a variety of suitable DDS modules available on AliExpress and eBay – see the parts list at the end of the article for sugestions.

Using a module like this takes a lot of the hard work out of the design.

By itself, the module will not do anything; it requires the power and control signals through the 10-pin header. It only took me a little while to design a control module for it. This has a microcontroller with a small display to show the frequency and a knob to set it. I kept the same display and appearance as the *Q Meter*, the earlier *Signal Generator* and associated projects.

Circuit details

The resulting circuit is shown in Fig.1. Microcontroller IC1 is a 28-pin DIP ATmega168 or ATmega328. Speed is not critical, so I am using the internal 8MHz RC clock source; no external crystal is needed. The display is the same SSD1306-based 128×64 pixel OLED screen as in my other designs, and the frequency is changed by a rotary encoder with a built-in pushbutton switch.

IC1 updates the display over a two-wire I^2C bus with the usual 4.7kΩ pull-up resistors. The rotary encoder terminals are pulled up by 4.7kΩ resistors, with 100nF and 470nF debouncing capacitors. The differing time constants make it easier for the micro to detect the encoder rotation reliably.

The *Generator* could run from any standard 5V plugpack, but as the current drain is not high, I decided to use two AA cells and a switch-mode boost converter to generate 4.4V DC. This boost converter is the same MCP1661 or MP1541 chip used in my *LC Meter* (*PE*, November 2023).

Why 4.4V instead of 5V? The resulting current consumption is lower, extending battery life. The AA cells should operate down to 1V each before the up-converter drops out. This voltage is set by the ratio of the 330kΩ and 120kΩ resistors to the feedback (FB) pin of REG1, which is maintained at 1.25V.

Since 1.25V × ([330kΩ ÷ 120kΩ] + 1) = 4.4V, the voltage at the cathode of D1 will increase until it reaches 4.4V, then REG1 will adjust its duty cycle to maintain that. The switch interrupting power from the battery to REG1 (S1) is onboard, making construction easier.

The AD9834 module is powered and controlled by IC1 via 10-pin header CON1. It has an onboard 75MHz oscillator, so the maximum output frequency (the Nyquist limit) is half that, ie, 37.5MHz. But it is best to operate it lower than that, so I chose a maximum of 25MHz.

As for the low end, the *Q Meter* needs a minimum frequency of 100kHz, but the module can go as low as 1Hz. I decided that 10Hz was a reasonable lower limit, spanning the full range of useful audio frequencies.

The resolution of the signal generator is 1Hz; pressing the pushbutton on the encoder toggles through step sizes of 1Hz, 10Hz, 100Hz, 1kHz, 10kHz, 100kHz and 1MHz. On power-up, the default step size is 1MHz.

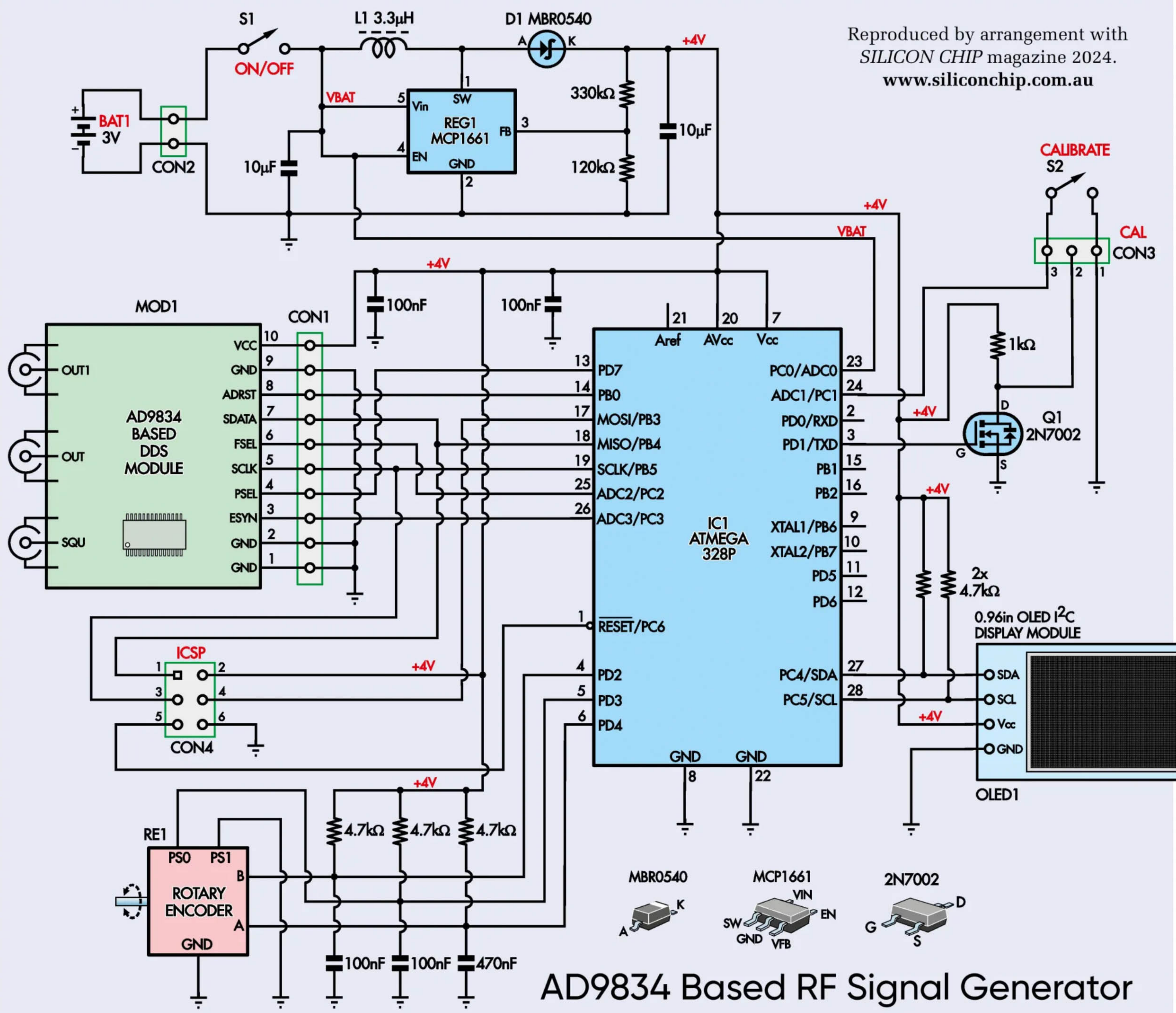
CON4 is a standard Atmel six-pin ICSP header that allows you to program IC1 in-circuit if fitted. There's also an optional serial debug interface at CON3; if you aren't using that, you can leave off MOSFET Q1 and its 1kΩ pull-up resistor. However, CON3 should be fitted as it is also used to trigger calibration when S2 is closed or its pins 1 and 3 are shorted.

Output frequency response

Once the firmware was working, I plotted the output level against frequency, shown as the red trace in Fig.2; two problems are apparent. The output was about -11dBm, which is too low, and it falls off rapidly above 18MHz.

The output level is set by one resistor, R2, which is 6.8kΩ on the supplied module. By changing this to 1.2kΩ, the output increased to near 0dBm over the flat part of the range, shown in green in Fig.2. The resistor on the module is an M1608/0603 size SMD type, but a larger M2012/0805 size resistor will also fit.

I measured the output power three ways, and they did not quite agree. The most reliable method is to measure the peak-to-peak voltage on an oscilloscope with an accurate 50Ω RF load


S1
ON/OFF
L1 3.3μH
D1 MBR0540
A
K
+4V
BAT1
3V
CON2
VBAT
Vin
SW
REG1
MCP1661
FB
EN
GND
10μF
330kΩ
120kΩ
10μF
CALIBRATE
S2
CAL
CON3
+4V
VBAT
MOD1
CON1
AD9834
BASED
DDS
MODULE
OUT1
OUT
SQU
VCC
GND
ADRST
SDATA
FSEL
SCLK
PSEL
ESYN
GND
GND
100nF
100nF
Aref
AVcc
Vcc
PD7
PB0
MOSI/PB3
MISO/PB4
SCLK/PB5
ADC2/PC2
ADC3/PC3
RESET/PC6
PD2
PD3
PD4
IC1
ATMEGA
328P
PC0/ADC0
ADC1/PC1
PD0/RXD
PD1/TXD
PB1
PB2
XTAL1/PB6
XTAL2/PB7
PD5
PD6
PC4/SDA
PC5/SCL
GND
GND
1kΩ
Q1
2N7002
D
G
S
2x
4.7kΩ
0.96in OLED I2C
DISPLAY MODULE
SDA
SCL
Vcc
GND
OLED1
ICSP
CON4
RE1
PS0
PS1
ROTARY
ENCODER
B
A
GND
4.7kΩ
4.7kΩ
4.7kΩ
100nF
100nF
470nF
MBR0540
MCP1661
VIN
EN
SW
GND
VFB
2N7002
AD9834 Based RF Signal Generator

Reproduced by arrangement with
SILICON CHIP magazine 2024.
www.siliconchip.com.au

Fig.1: the circuit is simple because the DDS signal generator is a prebuilt module that plugs into CON1. It's controlled by micro IC1, which monitors rotary encoder RE1 and displays the status on the OLED1 screen. Power comes from a pair of AA cells via boost converter REG1 that generates a steady 4.4V.

(how I plotted Fig.2). The other methods used the tinySA spectrum analyser and the Analog Devices AD8318 power meter. Those two methods gave values between 1dBm and 4dBm lower. This still leaves the problem of frequencies above 18MHz having a reduced level. If this is sufficient for your needs, no further modifications are needed. However, I decided that it was worthwhile to improve the frequency response.

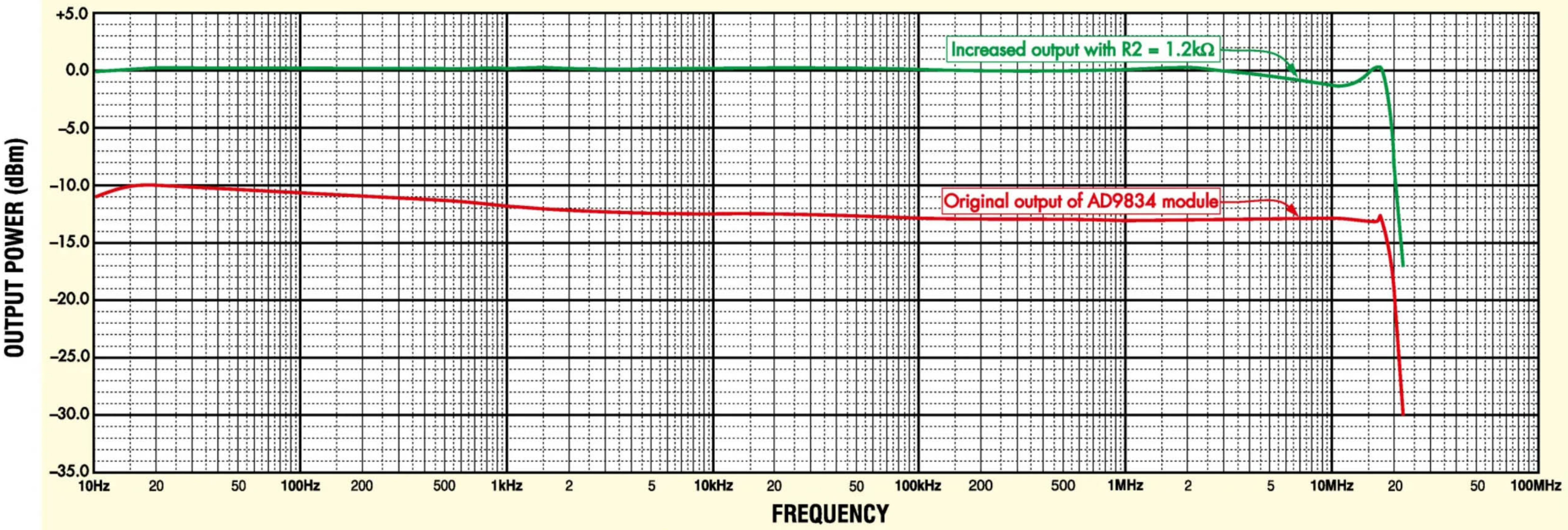


Fig.2: the output frequency response of the Signal Generator with the original resistor R2 (red) and new value (green).

If you look at my photos, you will see that the two outputs on the module each have a low-pass filter (LPF) consisting of three inductors and three capacitors. We can fix the drop-off by replacing L4-L6 and C7-C9 with different value components, giving a cutoff frequency of 35MHz.

The inductors are M2012/0805-size, and the capacitors are M1608/0603-size SMDs, but again, M2012/0805 size capacitors will fit. The new 5th-order Chebyshev LPF is shown in Fig.3. You will note that C8 is not needed in this topology. The new frequency response is shown in Fig.4. The other output can be left as-is, as it is unused.

Despite the 35MHz cutoff frequency, there is still a reduction at 25MHz due to the relatively low Q of the Coilcraft chip inductors I used; their rated Q factors are not high. At 25MHz, the 820nH inductor has a Q of 23, the 1.5µH has a Q of 10 and the 1.8µH has a Q of 15. These are not very impressive figures!

I tested some other SMD inductors that supposedly had a higher Q but they actually made the output level slightly lower. So the Coilcraft inductors are good if you can get them; the parts list includes some close alternatives that might be easier to get.

Harmonics

As with all DDS systems, the output is not pure, with multiple spurs. These are shown in Plots 1-3. Of the five frequencies I tested (5MHz, 10MHz, 15MHz, 20MHz and 25MHz), 15MHz and 25MHz give the purest output as they are one-fifth and one-third of the clock frequency. The only spurs are harmonics of the fundamental.

All others had multiple spurs, mostly more than 20dB down compared to the output frequency (the other two not shown are similar in appearance to Plot 2).

Housing it

I used a 105 × 75 × 40mm ABS enclosure with a clear lid, Altronics Cat H0321. An alternative is the Altronics Cat H0323 which is deeper at 55mm. Using the shallower H3021 case, there is only just enough spare room for the battery holder on the left side. The larger one has more room for the battery holder, allowing it to be attached to the bottom of the case.

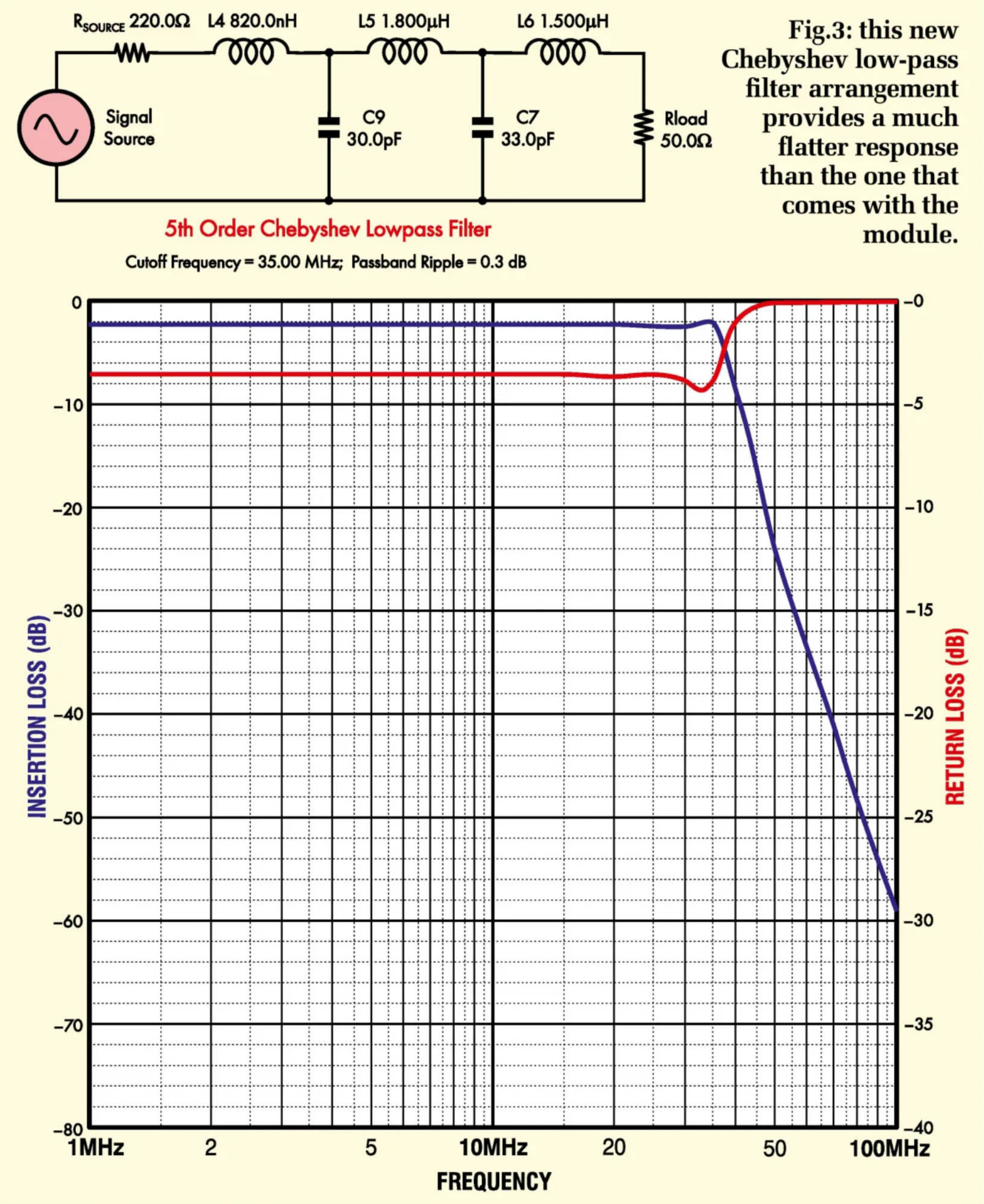


Fig.3: this new Chebyshev low-pass filter arrangement provides a much flatter response than the one that comes with the module.

Another advantage of the larger (H0323) case is that there is enough room to fit a potentiometer to allow you to adjust the output level from around -23dBm to 0dBm, to be detailed at the end of the article. Actually, you can fit a small (9mm or 10mm body size) potentiometer in the smaller (H0321) case, but using the larger case gives you more room and choices for that pot.

Apart from the AD9834 module, two circuit boards are used. One contains the control circuitry, and the other is the front panel with a cutout for the display and two holes for the switch and tuning shaft. This panel is a snug fit into the detent on the front panel, and it is held in place by the nut on the switch shaft.

Construction

The control board is built on a 59 × 65mm double-sided PCB coded CSE221001 that attaches to the clear lid by two screws in opposite corners. Both boards are available from the *PE PCB Service.*

Countersunk holes must be drilled in the clear panel for these screws – the best way to position the holes is to place the blank PCB inside the top cover hard up against the right side and use it as a template for drilling the holes for the switch, shaft encoder and two diagonal mounting holes.

After that, assemble the control board (see Fig.5), starting with the surface-mounting components. The resistors and capacitors are all 2.0 × 1.2mm, so they aren't too difficult to solder. However, the 5-pin SOT-23 chip (regulator REG1) requires some care due to its close pin spacing.

It really helps to have some flux paste to solder REG1. Put a little over the pads, then place the IC over them and tack one of the pins on the side with only two pins by loading a little solder on the clean tip of a soldering iron and touching it to both the pin and pad. Check that all the other pins are correctly located over their pads; if they aren't, re-heat the joint and gently nudge the regulator into position.

Once it's aligned, put a bit more solder on the iron and, after adding a little more flux paste, touch it to the three pins close to each other so that three good joints are formed. Check for bridges between the pins. If any have formed, add more flux paste and then use some solder wick to remove the excess solder. Finally, solder the last pin on the other side.

After that, all the through-hole components on the front side can be mounted. The OLED plugs into a 4-pin

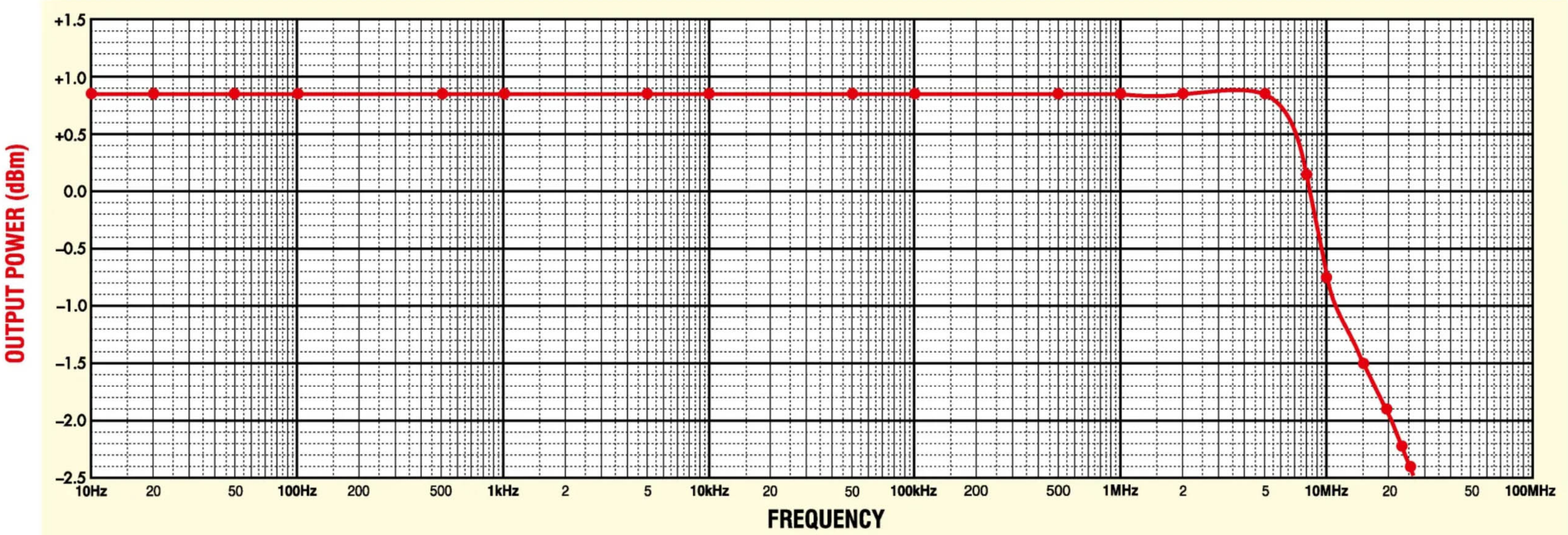


Fig.4: the measured response of the new filter. It isn't quite as good as estimated in Fig.3 due to the limited inductor Q values, but it's a vast improvement on the original. We calculate that the response to be significantly flatter than this using the specified components.

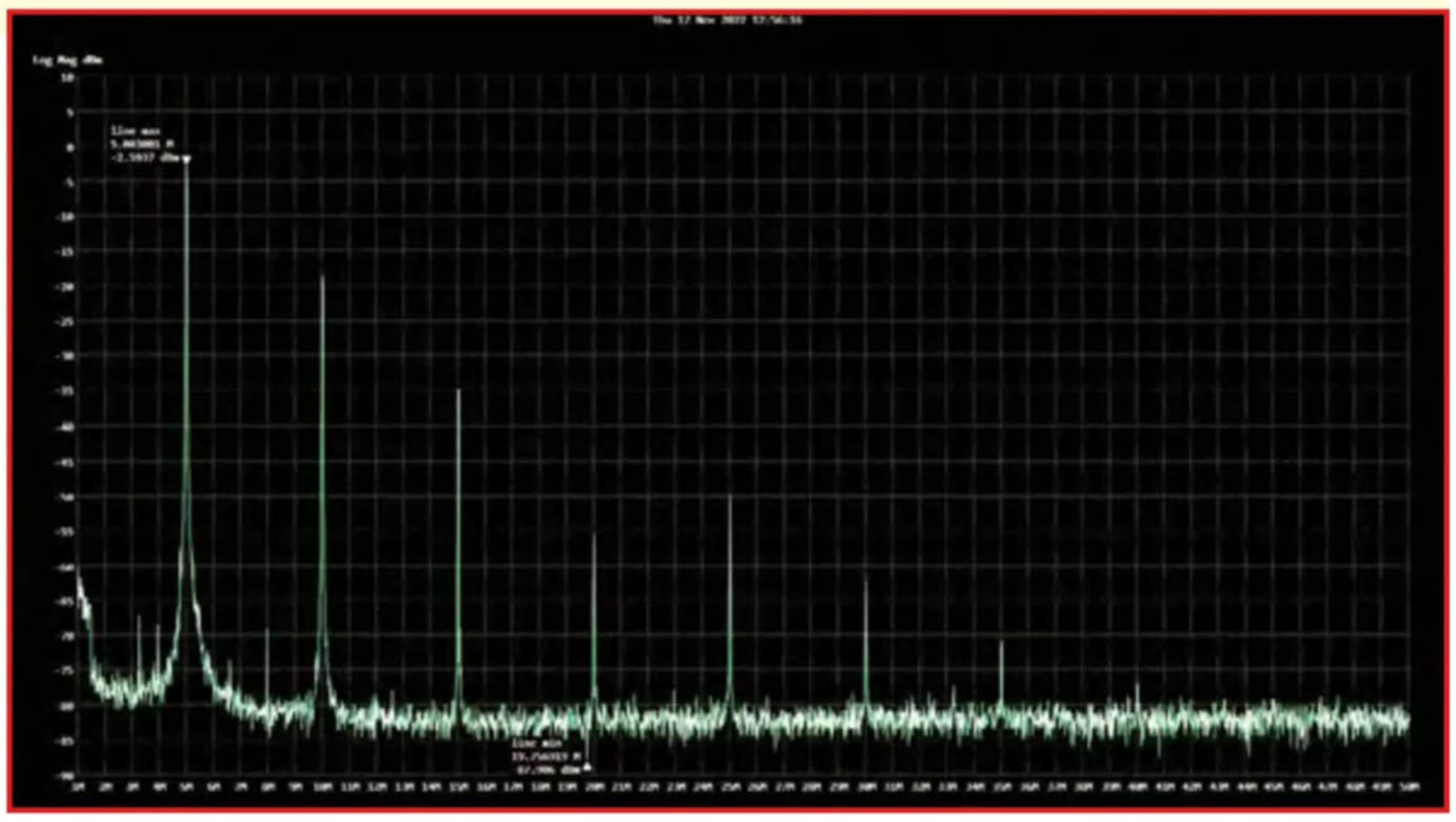
socket strip and is attached by 16mm screws through 8mm untapped spacers. Depending on the exact OLED screen used, the screws may need to be either 2mm or 2.5mm in diameter; most will accept M2 screws.

So the OLED sits at the right height, carefully slide off the plastic strip on the 4-pin header soldered to the OLED and cut the pins to suit the depth of the socket. Use a 28-pin DIL socket for the ATmega168/328. Finally, attach the three connectors on the underside of the board. Transistor Q1 and its associated 1kΩ resistor can be omitted if you don't need the serial debug feature.

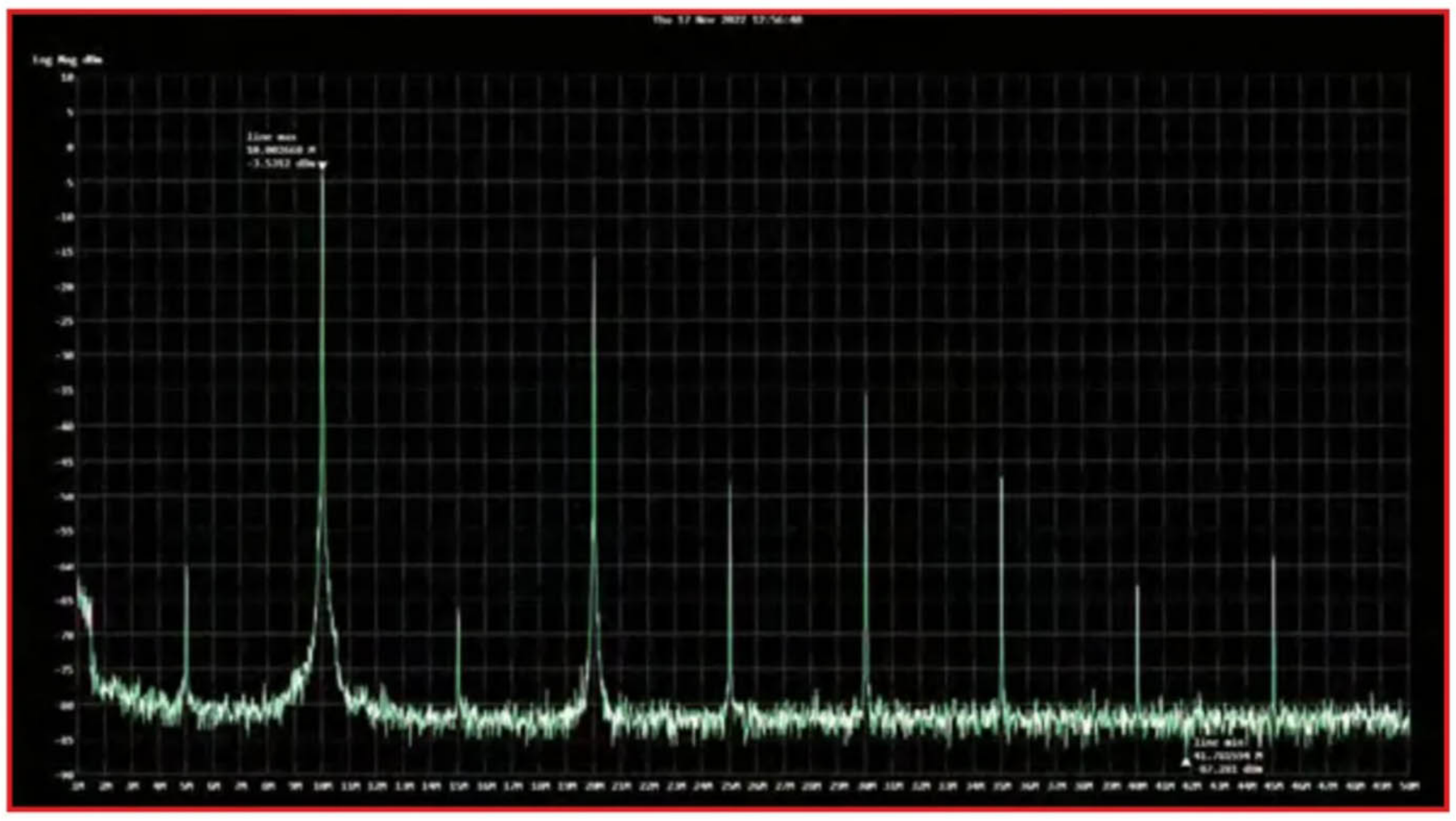
Assuming you have a blank chip, it is easiest to program it in-circuit using CON4. You will need an Atmel serial programmer; an Arduino can be used in this role. First, use the Arduino IDE to upload the ArduinoISP sample code to the Arduino to be used as a programmer. Next, wire up CON4 to the six-pin programming header on the Arduino, except for the RST signal on pin 5. Assuming you're using the Uno, pin 5 on CON4 goes to its D10 digital pin instead.

After that, you can use the free software AVRDUDE (Linux or Windows command-line) or AVRDUDESS (Windows GUI) to upload the HEX file available from the April 2024 page of the *PE* website: **https://bit.ly/pe-downloads**

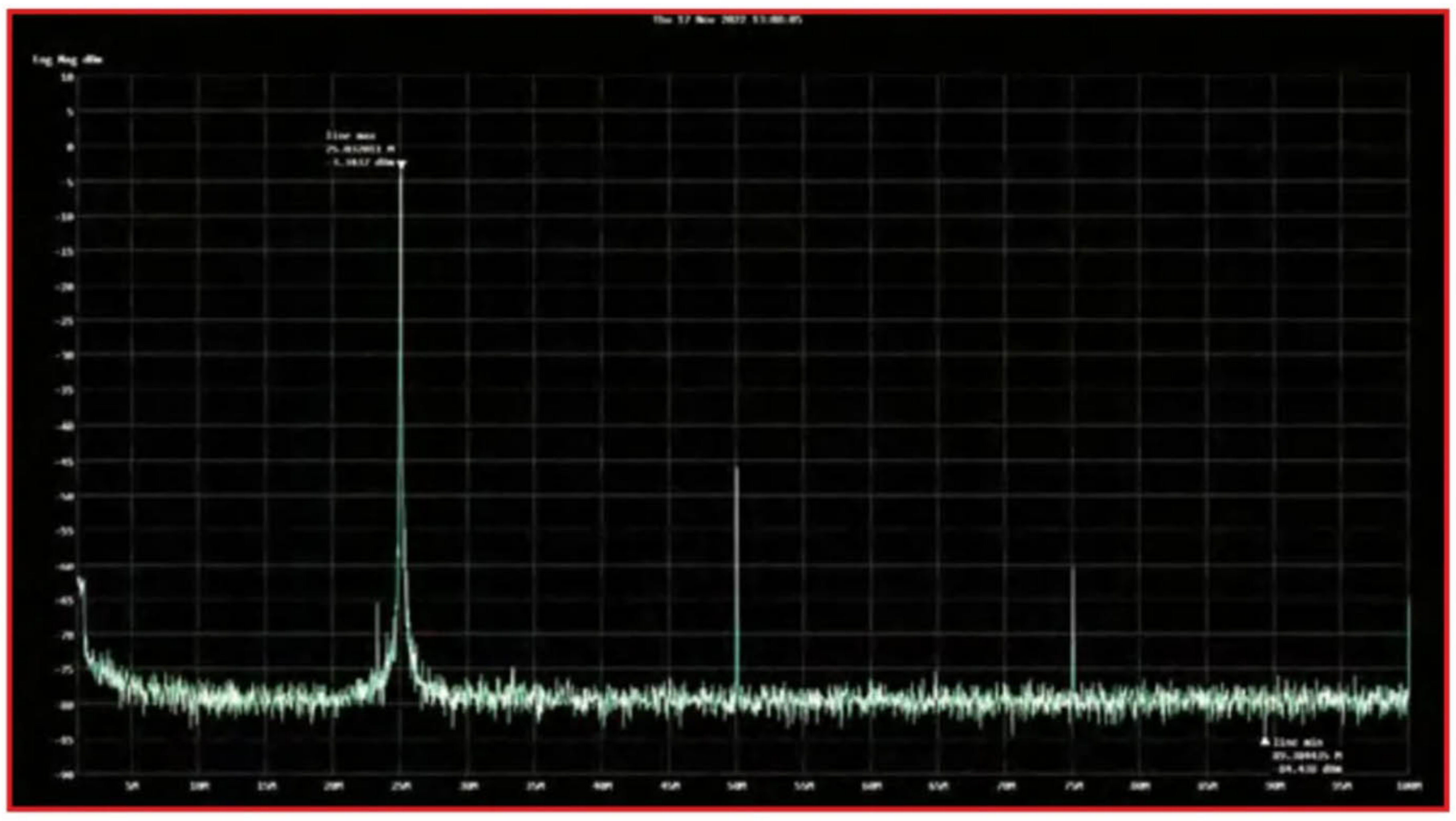
Use the 'Arduino' as the programmer and 19,200 as the baud rate. Make sure you select the correct COM port (the one the Arduino programmer board is using) and the target chip (ATmega168, ATmega328 etc).



Plot 1
5MHz



Plot 2
10MHz



Plot 3
25MHz

Modifying the DDS module

First, desolder and remove the small SMDs labelled L4-L6 and C7-C9 from the board. You can do this with a standard iron by grabbing one component at a time with some reasonably solid tweezers, then alternately heating one side and the other while pulling up gently until the part lifts off the board. It usually helps to melt a little extra tin-lead solder into the pad on each side before doing this.

Once the parts are off the board, squirt a small blob of flux paste onto each pad, place some solder wick on

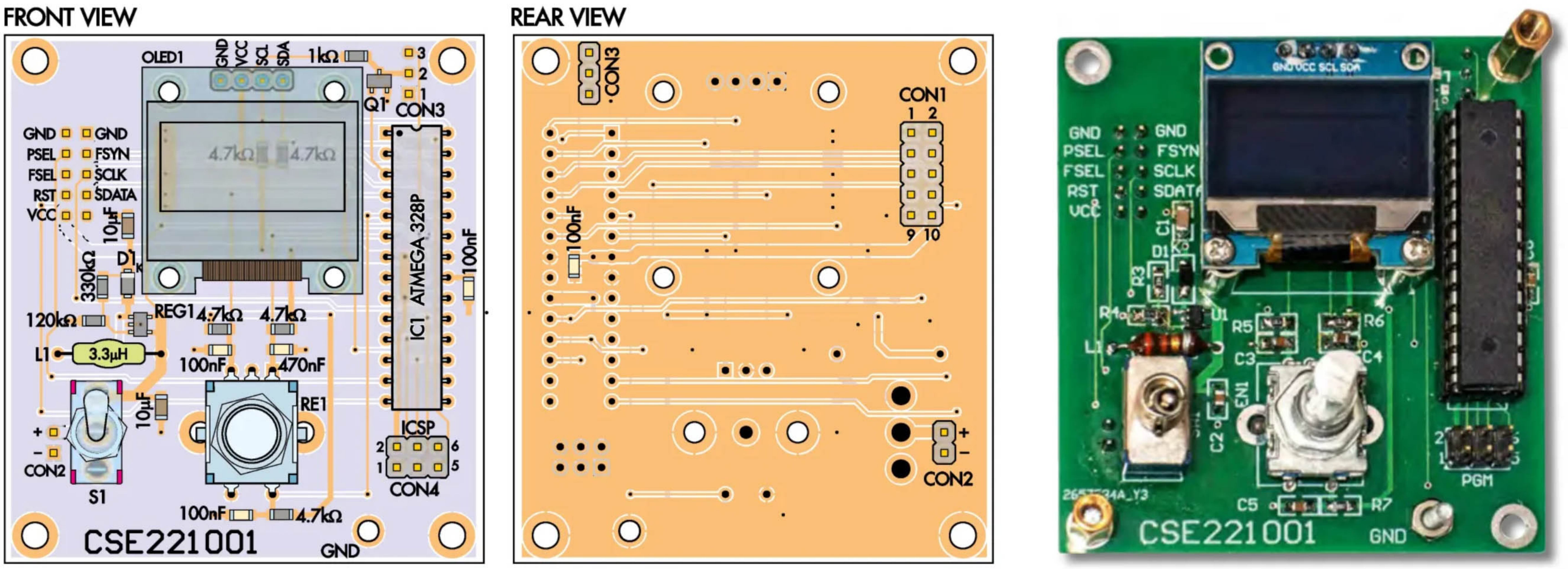


Fig.5: fit the components to the control board as shown here; note that there are two 4.7kΩ resistors under the OLED screen and a 100nF capacitor on the underside of the PCB.

top, press down with the iron, and, when it's hot enough, slide it off the pad. That should remove all but a very thin layer of solder. Clean up the flux residue with some flux cleaner or pure alcohol and a lint-free cloth or cotton bud.

You can then install all the new components: **L4 = 820nH, L5 = 1.8µH, L6 = 1.5µH, C7 = 33pF and C9 = 30pF**. Do not install a new capacitor on the pads for C8. Note that one of the specified inductors has an open side which should face towards the PCB while the other inductors and the capacitors can be fitted in any orientation.

Making the cable

A short 10-pin flat ribbon cable with IDC connectors at each end joins the two modules. Crimp the IDC connectors as shown in Fig.6; if in doubt, check the photos. You can use a vice to close down the connector on the flat cable, making sure that it is exactly square, although it's better to use a dedicated IDC crimping tool (eg, Altronics T1540).

There is no room for the strain relief clips on the connectors, so leave them off if supplied.

Testing

For initial testing, before assembling it into the case, connect the battery and switch it on. The OLED should come up with an initial message showing the version number. After two seconds, the display will show the frequency, step size and battery voltage. The default frequency on power-up is 10MHz, and the step size is 1MHz.

Check the VCC voltage at pin 7 or 20 of IC1; it should be close to 4V. You can use the labelled pad near the bottom edge of the PCB as a GND reference.

Rotating the knob should increase or decrease the frequency. Depending on the shaft encoder, it may operate backwards. If that is the case, then simply plug a jumper on the programming header between pin 4 and pin 6 of CON4. If you haven't fitted the header, you can do it now or solder a short component lead off-cut between those pins.

The firmware reads the level on digital input PORTB.3, which determines the encoder direction sensing. Adding a jumper between pins 4 and 6 of CON4 pulls that pin to GND.

If all is well, connect the AD9834 module, being careful with the orientation of the flat cable, ensuring that pin 1 goes to GND at both ends. A green LED on that module should light up when power is applied.

Check the output on the two SMA connectors with an RF power meter or oscilloscope. The output of the LPF requires 50Ω termination; without it, there may be some distortion of the output waveform.

Final assembly

Attach the control board to the transparent lid by two screws on opposite corners with 12mm-long M3 tapped spacers, into the countersunk holes you made earlier.

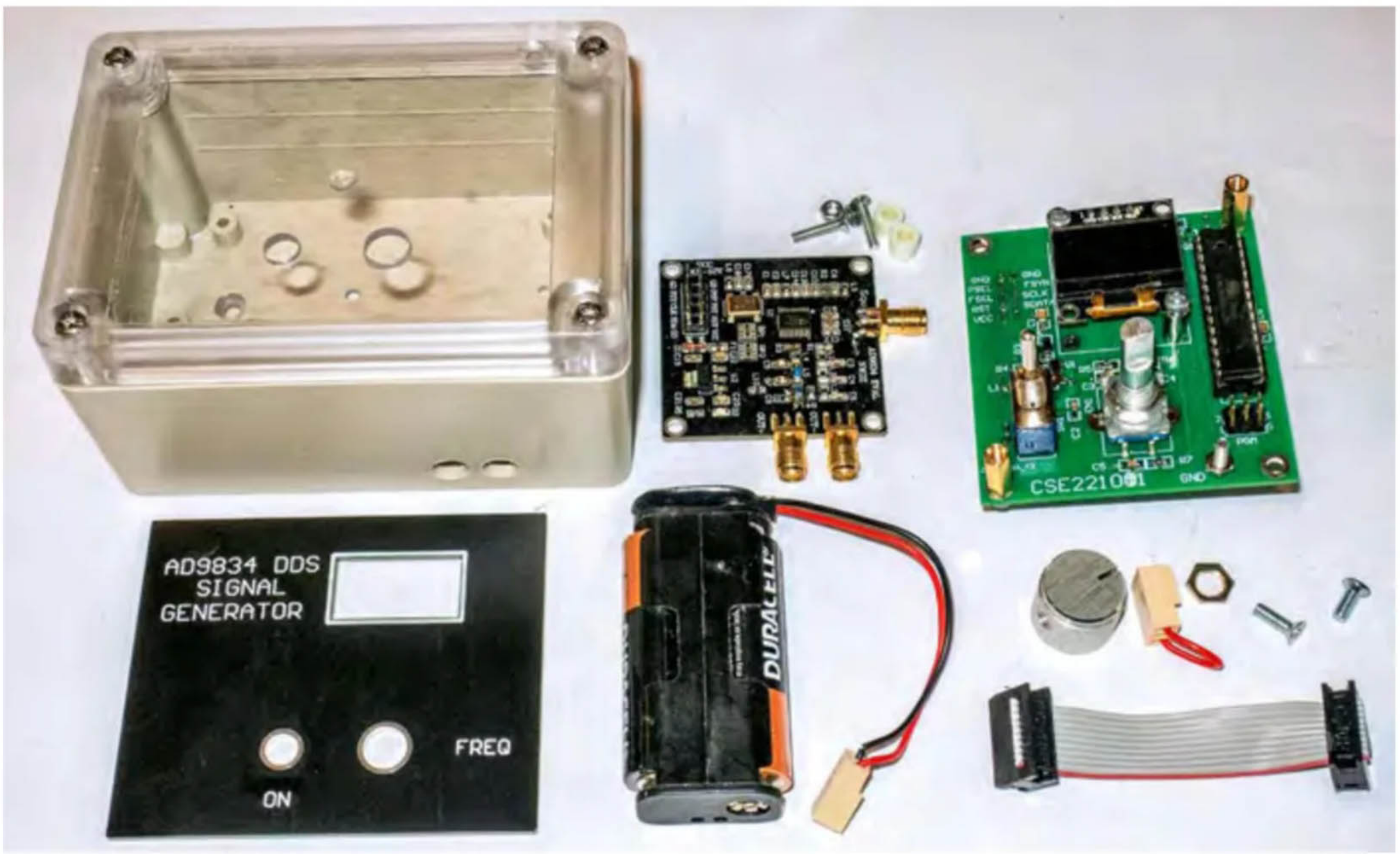
The AD9834 module attaches to the bottom of the case with M2/M2.5 × 12mm countersunk (CSK) screws and nuts, plus 5mm untapped spacers.

First, two holes need to be drilled in the side for the SMA connectors, as shown in Fig.7. The square wave output connector is not accessible and is not used in this design.

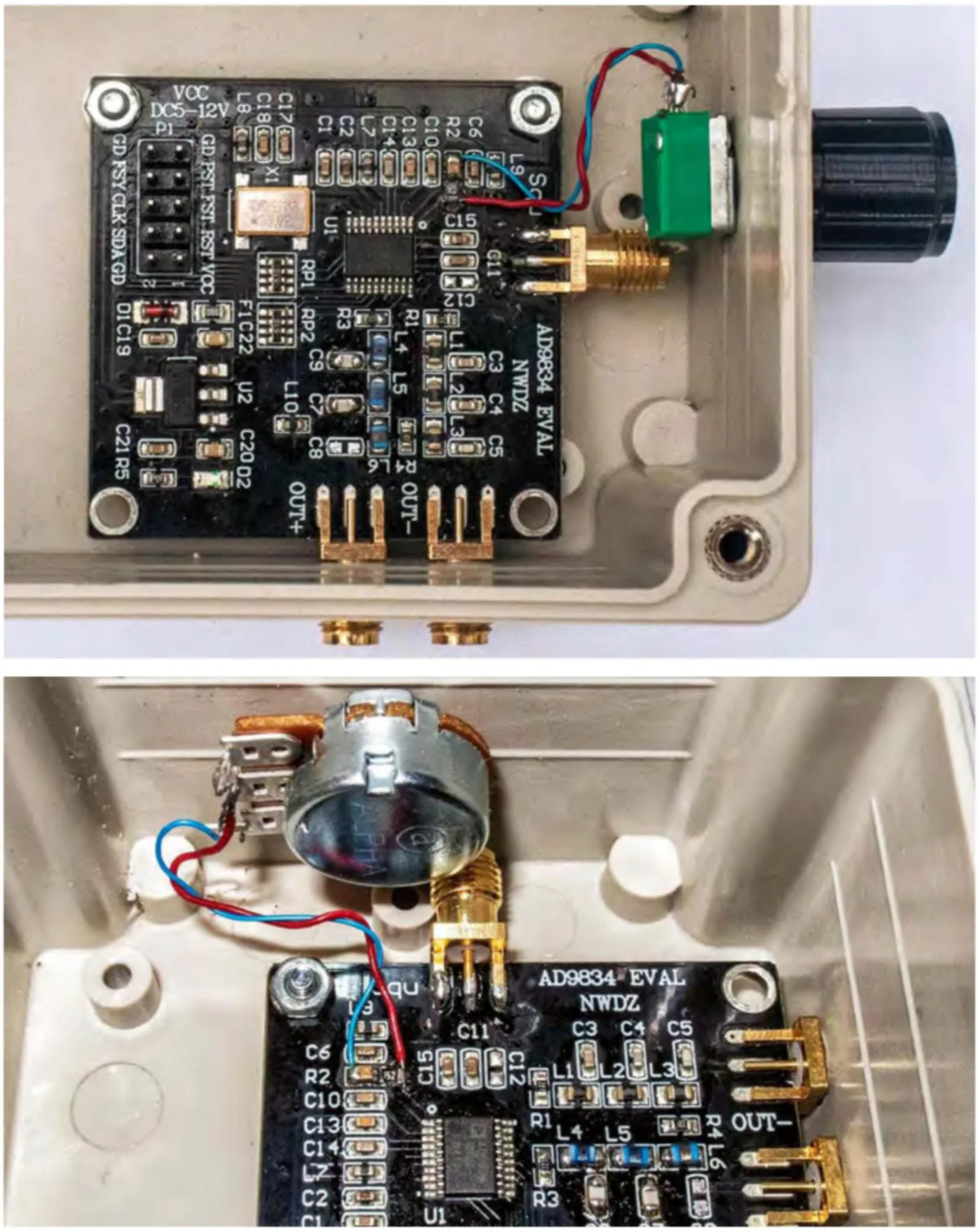
Next, slide in the module and use it as a template to mark the position of the two holes in the bottom. Drill these to 2.0mm or 2.5mm to suit your screws and then countersink them on the bottom.

Calibration

The output frequency accuracy depends on the exact frequency of the 75MHz oscillator on the module. I found the



This photo shows nearly all parts required to build the Basic RF Signal Generator, except for the replacement components for the DDS module (see the text above).



Inductors L4-L6 and capacitors C7-C9 have been replaced on the AD9834 module to provide a 35MHz cutoff frequency. The output level is adjusted by changing R2, which I replaced with a potentiometer.

error at 10MHz to be about 140Hz. This is of little importance for some applications, for exampe driving the *Q meter*. However, if you need it, there is a calibration procedure built in.

Set the frequency to precisely 10MHz and measure the output with a frequency counter. Turn on S2 or plug the jumper across CON3 and rotate the tuning knob until the readout on the counter is 10MHz ±1Hz, then press the knob. This sets a correction factor into an EEPROM which is read on power-up.

As there is no temperature compensation in the 75MHz crystal oscillator, you can expect this frequency to drift slightly, but it is likely to remain within ±20Hz at 10MHz. Recalibration may be needed from time to time as the crystal oscillator ages.

Battery life

With fresh AA alkaline cells, the input voltage is about 3.2V. The current drain starts at 80mA and increases as the battery voltage drops (because the boost regulator maintains a constant output voltage). By the time the battery drops to 2.7V, the current is about 95mA.

The best alkaline AA cells are 3000mAh, but that rating is for a light load. It has to be derated to 2000mAh or so at the expected current drain. This gives an expected operational life of about 20 hours.

Adjusting the output level

Depending on component values and settings, the *Q meter* can be fussy about its input signal level. Sometimes the 0dBm value is too high. We can use external attenuators, but this makes the setup rather complicated.

The output level is set by resistor R2 on the DDS module, so I thought why not use a potentiometer in its place? The wires to the potentiometer could pick up some noise that would amplitude-modulate the output. However, if the wires are short, that might not be a problem.

The adjacent photos shows how I did this on the prototype. I started by replacing R2 with a 100nF M2012/0805 SMD capacitor, providing noise filtering and a firmer base to the connecting wires. Connect a 1.2kΩ M2012/0805 SMD resistor to one end of this capacitor, then use short wires to connect a 50kΩ pot between the wiper and the anti-clockwise end of the track.

Mount this on the right-hand side of the enclosure so the wires are very short. Take great care in attaching the wires to the module to prevent any damage to the SMD connections.

With the maximum resistance, the output becomes about 22mV peak-to-peak, corresponding to about -29dBm. At minimum resistance, the output is close to 0dBm. I saw no evidence of noise pickup in the output signal.

If adding this output control, using the larger case (H0323) gives you more options; you could use a 16mm, 10mm or 9mm potentiometer. With the smaller case, you'll have to use a 9mm or 10mm potentiometer to have any chance of it fitting.

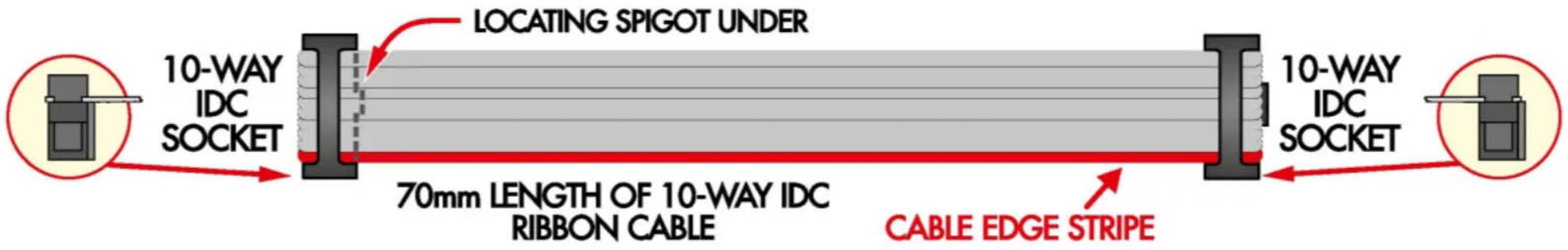


Fig.6: the ribbon cable is simple to make but ensure that the pins are fully pushed into the plastic housing, or you might end up with bad connections.

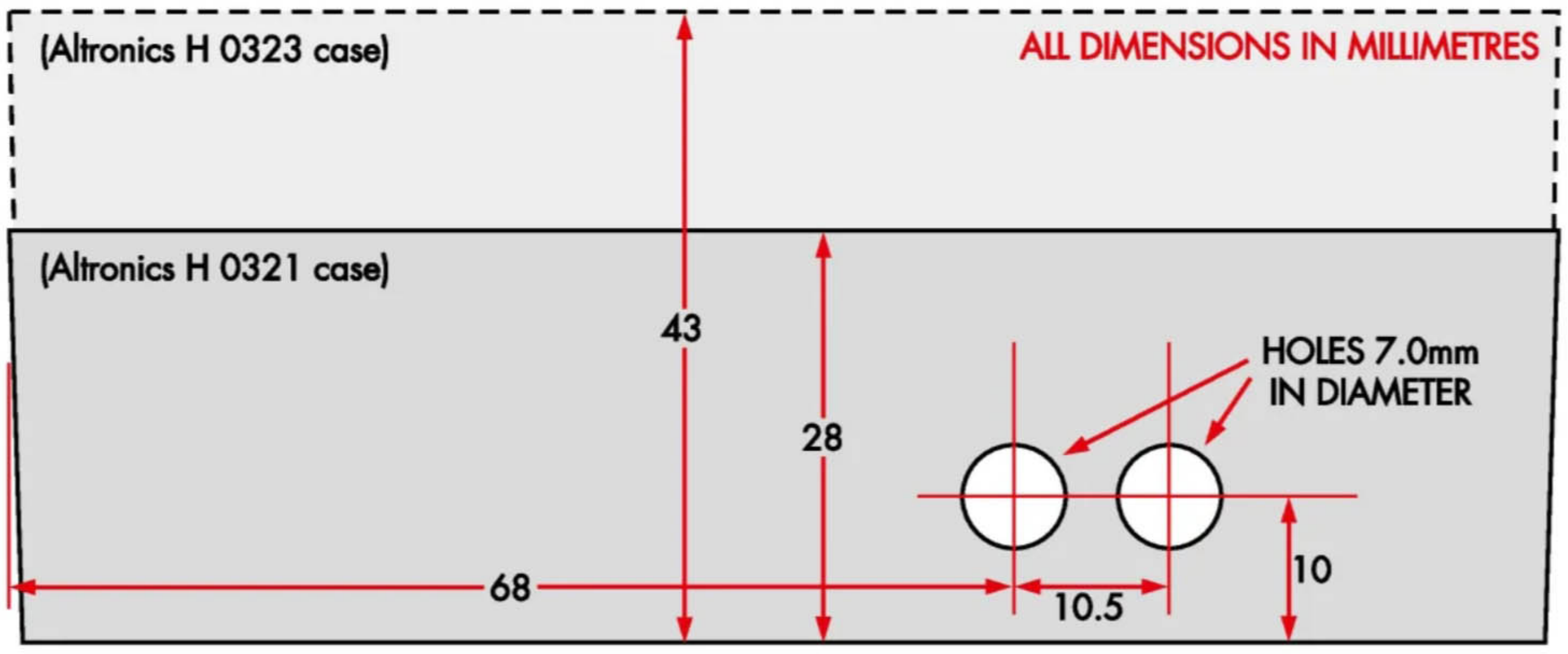


Fig.7: where to drill the holes in the side of the box for the SMA connectors.

PoLabs

For more information or
software download please visit

www.poscope.com/epe

PoKeys
Connect, Control

- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC

Stepper motor drivers
Drive

- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A

PoScope Mega1+
PoScope Mega50
Measure

- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

Parts List – Basic RF Signal Generator

1 double-sided PCB coded CSE221001, 59 × 65mm from the PE PCB Service
1 black PCB coded CSE220902B, 77.5 × 64mm, 1mm thick (front panel) from the PE PCB Service
1 0.96in OLED screen, SSD1306-compatible controller (OLED1) **[SC6176]**
1 AD9834-based RF DDS signal generator module (MOD1) (**see below**)
1 vertical-mount rotary encoder with integral pushbutton and 20mm-long shaft (RE1) **[SC5601]**
1 105×75×40mm or 105×75×55mm ABS case **[Altronics H0321 or H0323]**
1 3.3uH axial RF inductor (L1)
1 820nH SMD inductor, M2012/0805, Q = 100 @ 25MHz (L4 on MOD1) **[Coilcraft 0805HP-821XJRC or Vishay Dale IMC0805ERR82J01]** •
1 1.8µH SMD inductor, M2012/0805, Q ≈ 40 @ 25MHz (L5 on MOD1) **[Coilcraft 0805CS-182XJRC or Murata LQW21HN1R8J00L]** •
1 1.5µH SMD inductor, M2012/0805, Q ≈ 40 @ 25MHz (L6 on MOD1) **[Coilcraft 0805CS-152XJRC or Murata LQW21HN1R5J00L]** •
1 2×AA cell holder with flying leads (BAT1)
2 AA alkaline cells
1 2×5 pin header (CON1)
1 2-pin polarised header with matching plug and pins (CON2)
1 3-pin polarised header with matching plug and pins (CON3)
1 2×3 pin header (CON4; **optional**; for in-circuit programming of IC1)
1 jumper shunt (**optional**; to set the direction of RE1)
1 4-pin female header (for OLED1)
2 SPDT chassis-mount toggle switches with solder tags (S1 and S2; **S2 is optional**, for calibration)
1 28-pin DIL IC socket (for IC1)
2 10-way IDC crimp sockets

Cable and hardware
1 knob to suit RE1
2 M3 × 12mm tapped spacers
2 M3 × 6mm panhead machine screws
2 M3 × 6mm countersunk head machine screws
2 M2.5 or M2 × 12mm countersunk head machine screws
2 M2.5 or M2 × 16mm panhead machine screws
4 M2.5 or M2 hex nuts
2 3mm ID, 8mm long untapped spacers
2 3mm ID, 5mm long untapped spacers
1 70mm length of 10-way ribbon cable
1 double-sided foam tape pad or strips (to secure the cell holder)

Semiconductors
1 ATmega168P or ATmega328P programmed with CSE22100A.HEX (IC1)
1 MCP1661T-E/OT or MP1541DJ-LF-P integrated high-voltage boost regulator, SOT-23-5 (REG1)
1 2N7002 N-channel signal MOSFET, SOT-23 (Q1; **optional**, debug interface)
1 MBR0540 50V 500mA schottky diode, SOD-123 (D1)

Capacitors *(all SMD ceramic, M2012/0805 size, unless noted)*
2 10µF 6.3V X5R or X7R
1 470nF 6.3V X7R
4 100nF 50V X7R
1 33pF 50V C0G/NP0 (C7 on MOD1) •
1 30pF 50V C0G/NP0 (C9 on MOD1) •
• *replacement parts for the AD9834 DDS module*

Resistors *(all 1% SMD M2012/0805 size, unless noted)*
1 330kΩ 1 120kΩ 5 4.7kΩ 1 1kΩ (**optional**, debug interface)

Additional parts for adjustable output level
1 100nF 50V X7R SMD M2012/0805 size ceramic capacitor
1 1.2kΩ 1% SMD M2012/0805 size resistor
1 50kΩ chassis-mounting single-gang linear potentiometer **[Altronics R2245 or Jaycar RP8516]**
1 short length of light-duty figure-8 wire (eg, stripped from ribbon cable)

AD9834 module

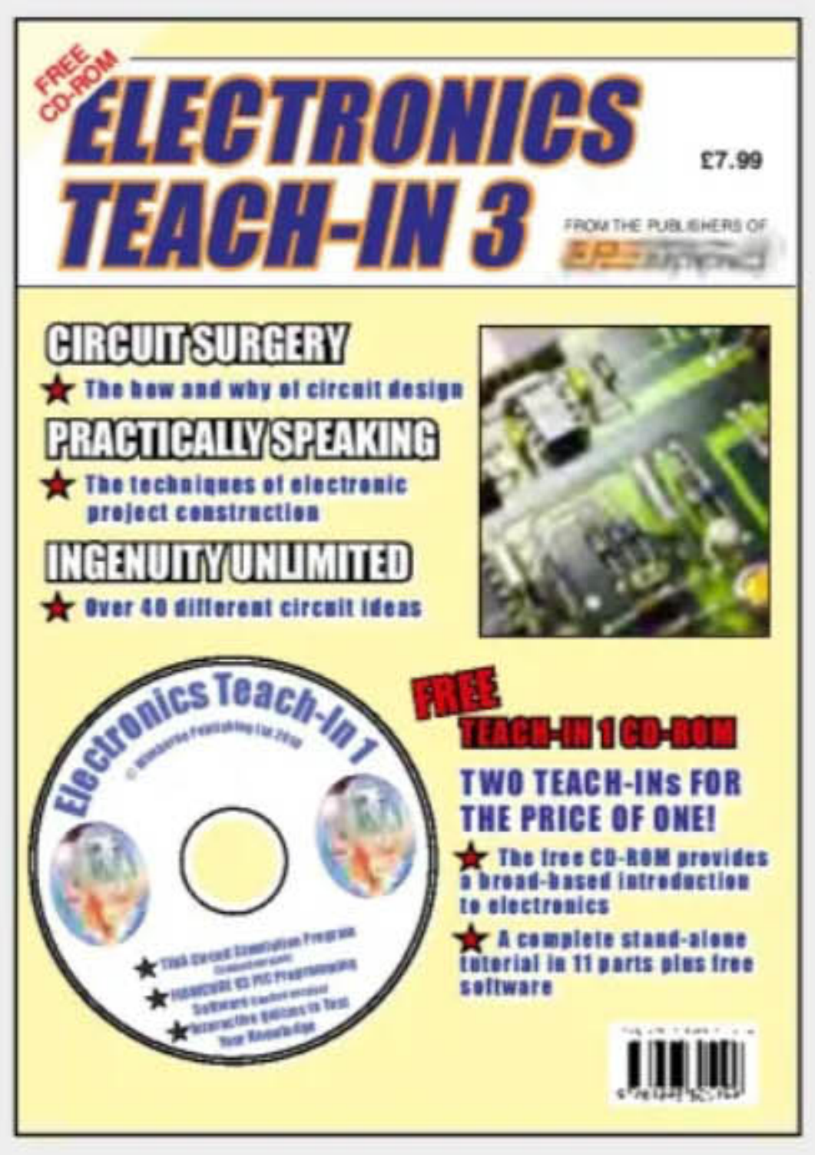
Annoyingly, my original source for this module has now stopped selling it. When sourcing the part, check that the pin next to VCC is FST, and that there are four components next to the SOT-223 regulator in the corner of the board. The following international eBay suppliers offer it.

www.ebay.com.au/itm/254760896955
www.ebay.com.au/itm/264914563563
www.ebay.com.au/itm/145517726380
www.ebay.com.au/itm/184507223305
www.ebay.com.au/itm/224210854323

ETI BUNDLE (1) Teach-In 3, 4 and 5 – all on CD-ROM – only £18.95

ELECTRONICS TEACH-IN 3 – CD-ROM

Mike & Richard Tooley



The three sections of the *Teach-In 3* CD-ROM cover a huge range of subjects that will interest everyone involved in electronics – from newcomers to the hobby and students to experienced constructors and professionals.

The first section (80 pages) is dedicated to *Circuit Surgery*, *EPE/PE's* regular clinic dealing with readers' queries on circuit design problems – from voltage regulation to using SPICE circuit simulation software.

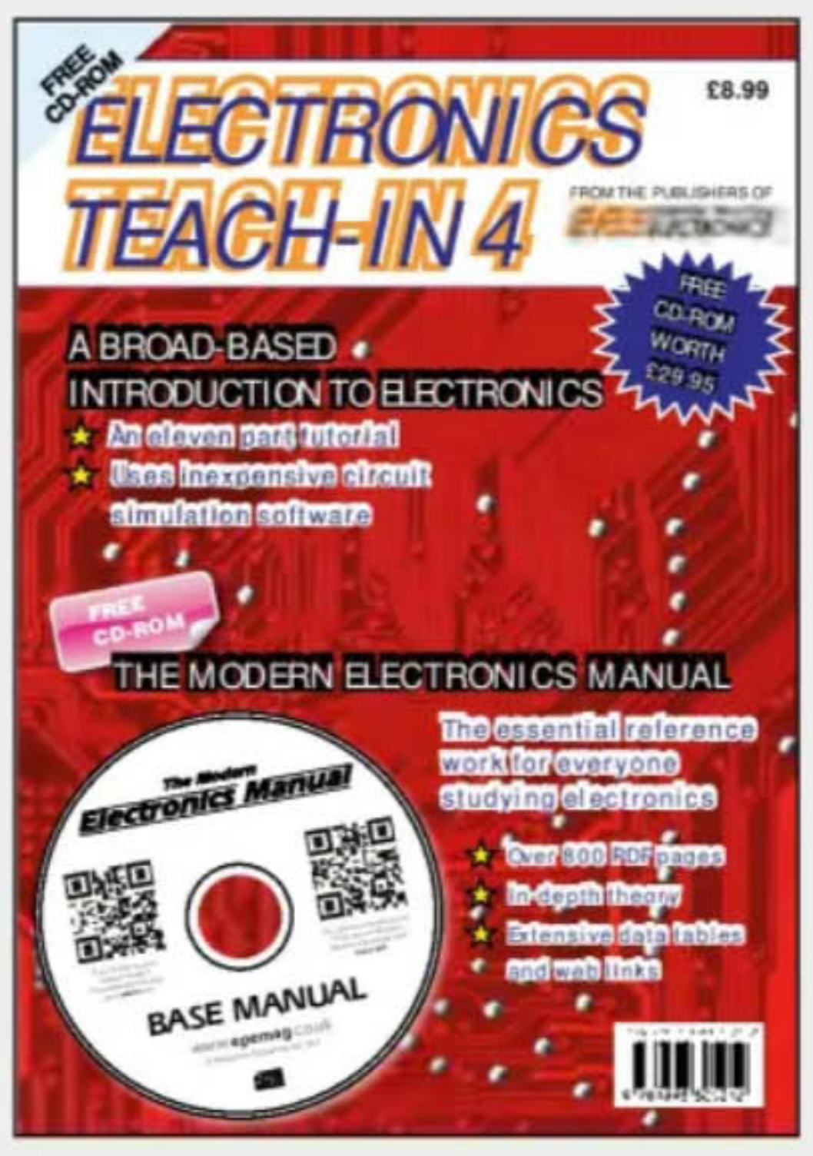
The second section – *Practically Speaking* – covers hands-on aspects of electronics construction. Again, a whole range of subjects, from soldering to avoiding problems with static electricity and identifying components is covered. Finally, our collection of *Ingenuity Unlimited* circuits provides over 40 circuit designs submitted by readers.

The CD-ROM also contains the complete *Electronics Teach-In 1* book, which provides a broad-based introduction to electronics in PDF form, plus interactive quizzes to test your knowledge and TINA circuit simulation software (a limited version – plus a specially written TINA Tutorial).

The *Teach-In 1* series covers everything from electric current through to microprocessors and microcontrollers, and each part includes demonstration circuits to build on breadboards or to simulate on your PC.

ELECTRONICS TEACH-IN 4 – CD-ROM
A BROAD-BASED INTRODUCTION TO ELECTRONICS

Mike & Richard Tooley


ELECTRONICS TEACH-IN 4
A BROAD-BASED INTRODUCTION TO ELECTRONICS
THE MODERN ELECTRONICS MANUAL

The *Teach-In 4* CD-ROM covers three of the most important electronics units that are currently studied in many schools and colleges. These include, Edexcel BTEC level 2 awards and the electronics units of the Diploma in Engineering, Level 2.

The CD-ROM also contains the full *Modern Electronics Manual*, worth £29.95. **The Manual** contains over 800 pages of electronics theory, projects, data, assembly instructions and web links.

A package of exceptional value that will appeal to anyone interested in learning about electronics – hobbyists, students or professionals.

ELECTRONICS TEACH-IN 5 – CD-ROM
JUMP START

Mike & Richard Tooley

ELECTRONICS TEACH-IN 5
JUMP START
PRACTICALLY SPEAKING
PIC 'N' MIX
TEACH-IN 2
TWO TEACH-INS FOR THE PRICE OF ONE!

15 design and build circuit projects for newcomers or those following courses in school and colleges.

The projects are: ■ Moisture Detector ■ Quiz Machine ■ Battery Voltage Checker ■ Solar-Powered Charger ■ Versatile Theft Alarm ■ Spooky Circuits ■ Frost Alarm ■ Mini Christmas Lights ■ iPod Speaker ■ Logic Probe ■ DC Motor Controller ■ Egg Timer ■ Signal Injector Probe ■ Simple Radio Receiver ■ Temperature Alarm.

PLUS

PIC'n' Mix – starting out with the popular range of PIC microcontrollers and *Practically Speaking* – tips and techniques for project construction.

The CD-ROM also contains:

- ■ Complete *Teach-In 2* book, a practical introduction to PIC microprocessors
- ■ MikroElektronika, Microchip and L-Tek PoScope software.

Three Teach-ins for the great price of £18.95

PLUS you also get the contents of the free CD-ROM from each issue...

... so that's another TWO Teach-Ins and The Full Modern Electronics Manual!

What a Bargain!!

ETI BUNDLE (2) Teach-In 6, 7 and 8 – all on CD-ROM – only £18.95

ELECTRONICS TEACH-IN 6 – CD-ROM
A COMPREHENSIVE GUIDE TO RASPBERRY Pi

Mike & Richard Tooley

Teach-In 6 contains an exciting series of articles that provides a complete introduction to the Raspberry Pi, the low cost computer that has taken the education and computing world by storm.

This latest book in our *Teach-In* series will appeal to electronics enthusiasts and computer buffs who want to get to grips with the Raspberry Pi.

Teach-In 6 is for anyone searching for ideas to use their Pi, or who has an idea for a project but doesn't know how to turn it into reality. This book will prove invaluable for anyone fascinated by the revolutionary Pi. It covers:

- ■ Pi programming
- ■ Pi hardware
- ■ Pi communications
- ■ Pi Projects
- ■ Pi Class
- ■ Python Quickstart
- ■ Pi World
- ■ ...and much more!

The *Teach-In 6* CD-ROM also contains all the necessary software for the series, so that readers and circuit designers can get started quickly and easily with the projects and ideas covered.

ELECTRONICS TEACH-IN 6
RASPBERRY Pi
A COMPREHENSIVE GUIDE TO RASPBERRY Pi

ELECTRONICS TEACH-IN 7 – CD-ROM
DISCRETE LINEAR CIRCUIT DESIGN

Mike & Richard Tooley

Teach-In 7 is a complete introduction to the design of analogue electronic circuits. It is ideal for everyone interested in electronics as a hobby and for those studying technology at schools and colleges. The CD-ROM also contains all the circuit software for the course, plus demo CAD software for use with the *Teach-In* series.

- ■ Discrete Linear Circuit Design
- ■ Understand linear circuit design
- ■ Learn with 'TINA' – modern CAD software
- ■ Design simple, but elegant circuits
- ■ Five projects to build:
  - i) Pre-amp
  - ii) Headphone Amp
  - iii) Tone Control
  - iv) VU-meter
  - v) High Performance Audio Power Amp.

PLUS

Audio Out – an analogue expert's take on specialist circuits

Practically Speaking – the techniques of project building.

ELECTRONICS TEACH-IN 7
DISCRETE LINEAR CIRCUIT DESIGN

ELECTRONICS TEACH-IN 8 – CD-ROM
INTRODUCING THE ARDUINO

Mike & Richard Tooley

Hardware: learn about components and circuits
Programming: powerful integrated development system
Microcontrollers: understand control operations
Communications: connect to PCs and other Arduinos.

Teach-In 8 is an exciting series designed for electronics enthusiasts who want to get to grips with the inexpensive, popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. It will provide a one-stop source of ideas and practical information.

The Arduino offers a truly effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle through wireless or the Internet.

Teach-In 8 is based around a series of practical projects with plenty of information for customisation.

This book also includes *PIC n' Mix*: 'PICs and the PICkit 3 – A Beginners guide' by Mike O'Keefe and *Circuit Surgery* by Ian Bell – 'State Machines part 1 and 2'.

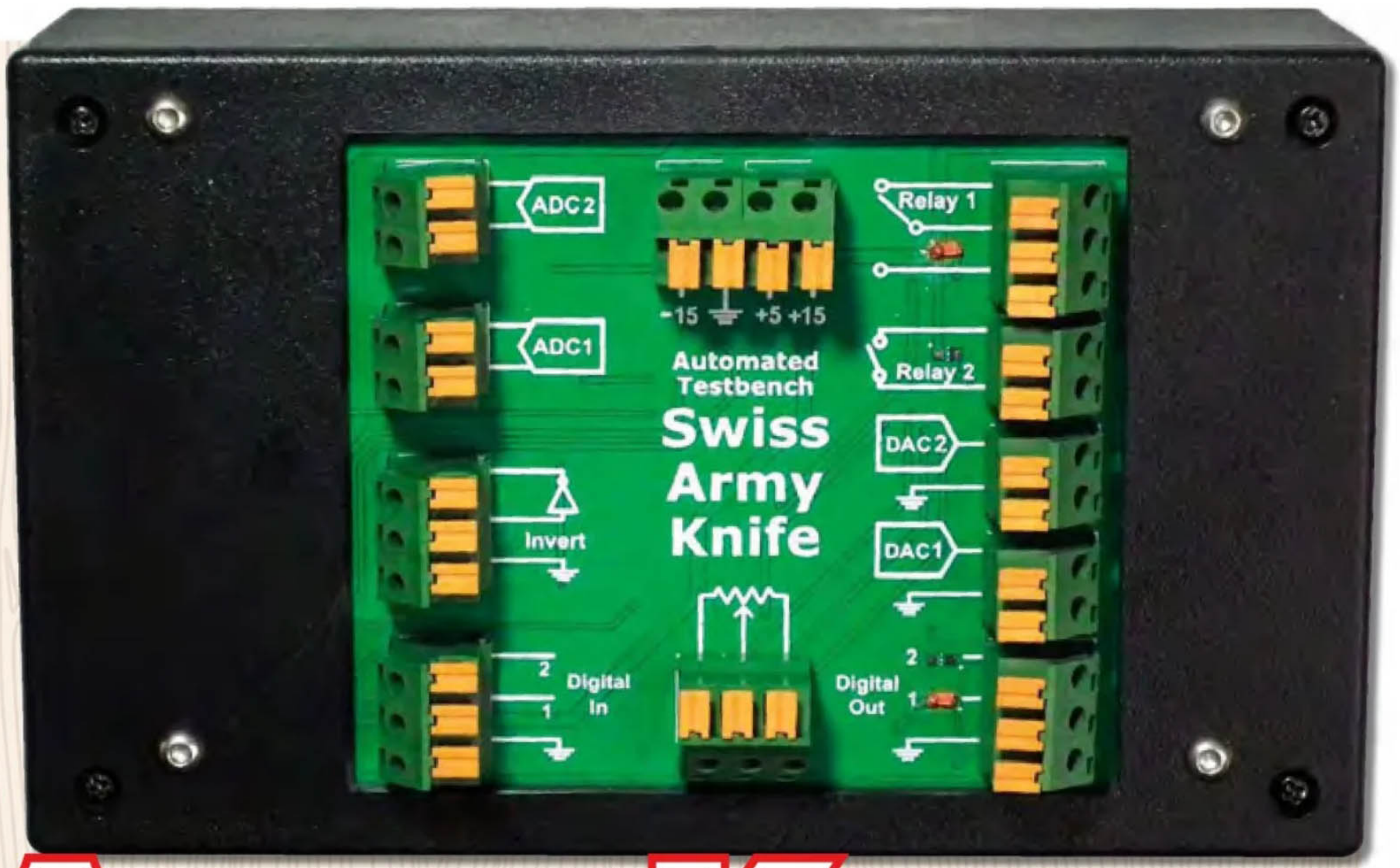
The CD-ROM includes the files for:

- ■ *Teach-In 8*
- ■ Microchip MPLAB IDE XC8 8-bit compiler
- ■ PICkit 3 User Guide
- ■ Lab-Nation Smartscope software.

ELECTRONICS TEACH-IN 8
INTRODUCING THE ARDUINO

ORDER YOUR BUNDLE TODAY!
JUST CALL 01202 880299 – OR VISIT www.electronpublishing.com

This handy device can provide test voltages, test signals, vary a resistance, switch a component in or out of circuit and even measure some voltages. It ties into automation software so it performs tests automatically and records input and output values for analysis.



Swiss Army Knife

An Automated Test Bench by Richard Palmer

When testing something on the bench, I often need to fish around in the parts drawer for some control component, like a switch or a pot. That's so I can test some circuitry across a range of voltages, with different component values or with some element in and out of circuit. I'm usually also measuring the impact of changes at one or two places in the circuit.

It's remarkable how often I reach for the same components: a switch, a 100kΩ pot, a sinewave generator and a 0-10V control voltage source being among the most frequent.

A collection of these most-used elements would be like a 'Swiss Army Knife' for the test bench. Most pocket knives don't pretend to have all the tools you'll ever need, or even the absolute best of each kind of tool. Still, they offer a set of robust, basic tools that will get the job done when the perfect tool isn't at hand or isn't needed.

The cost and complexity of the project have been kept down by controlling it via Wi-Fi using a web interface rather than an LCD screen. That also means it can be teamed up with test automation software, such as TestController, to automate many test bench tasks.

Features and performance

Pocket knives range from a single blade to monsters with more than thirty functions. We've settled on nine functions for this project, and focused on making them simple to use while designing them to tolerate moderate abuse.

The input and output connections are made with spring-loaded or cam-operated terminals and multiple ground connections are provided.

Two 16-bit analogue inputs with over-voltage protection can measure ±10V DC to within a few millivolts with excellent linearity. As long as both input terminals are kept within that range, it can measure differentials up to 20V. The input range can be extended by adding series resistance to the inputs.

The 0-10V DC analogue output has 256 steps of approximately 40mV (see Screen 1). While the accuracy isn't at the same level as the analogue inputs, 256 individual test values should be enough for most purposes.

The sinewave generator operates from 133Hz to 55kHz. The generator has two output voltages: 6V peak-to-peak (2.1V RMS) and 775mV peak-to-peak (0.27V RMS). The available frequencies are multiples of 133.33 Hz, and the software rounds settings down to the nearest available value.

Despite being driven by an 8-bit digital-to-analogue converter (DAC), the noise and distortion total less than 1.5% across the range (see Scope 1 and 2) after low-pass filtering. Major contributors to this are the sine generator DAC's voltage steps and a jump of several steps at the zero-crossing point. These artefacts are much less visible on the high output range, making that the range of choice.

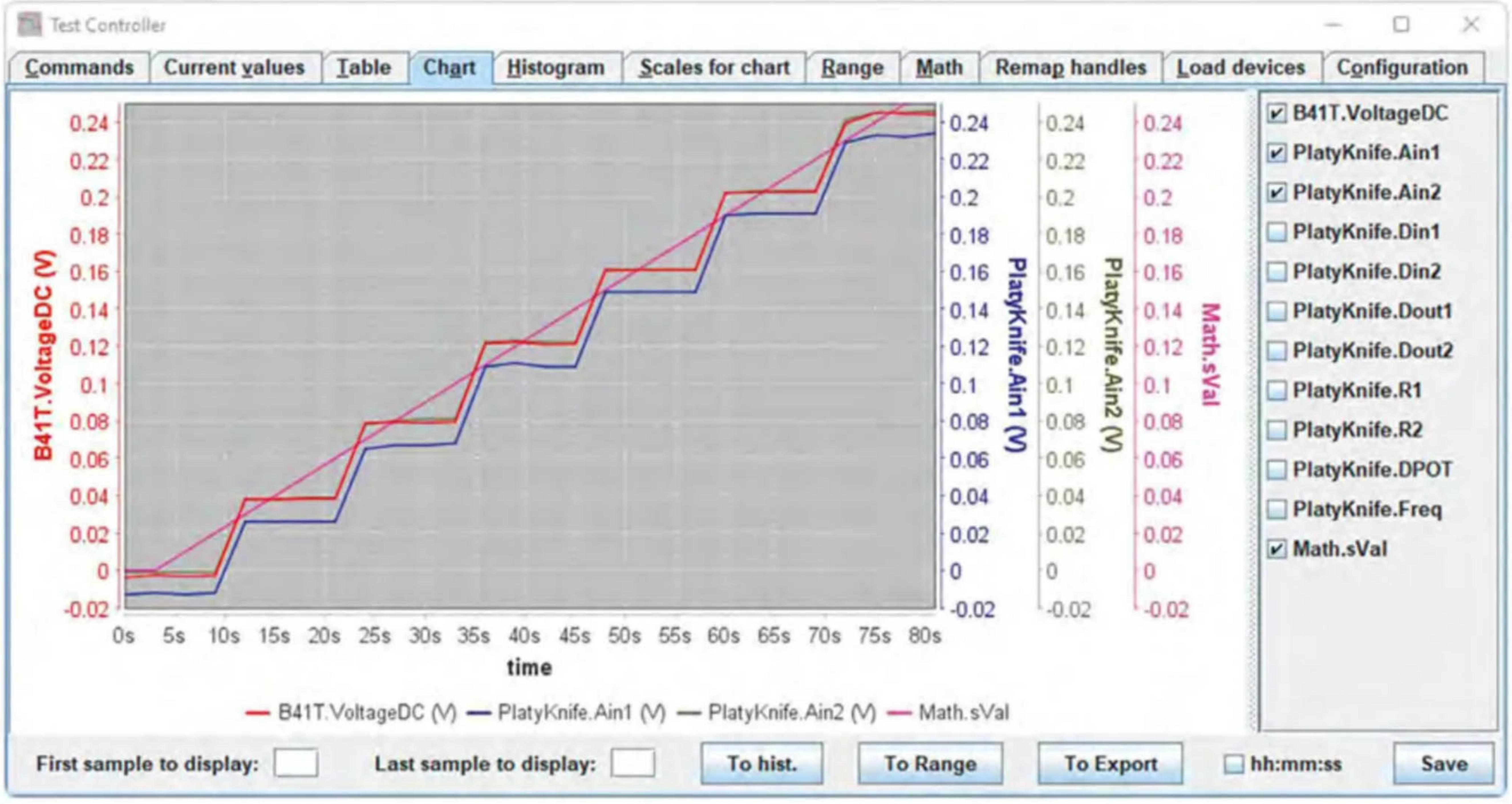
When finer voltage control is desired, the sinewave generator can be teamed up with the digital pot to provide 256 voltage attenuation steps for either of the basic output voltages.

A general-purpose op amp based inverter is included to provide additional flexibility in handling negative input or output voltages.

We've included two different relays: RLY1 is a 350mA SPST reed relay, useful for switching signals, while RLY2

Features and Specifications

- 256-step, 0-10V output (from a DAC)
- 133Hz-55kHz sinewave generator
- Two ±10V fully-differential analogue inputs (16-bit ADC)
- Analogue inverter with ±10V input and output ranges
- Two 3.3V digital outputs
- Two 5V-tolerant digital inputs
- 100kΩ digital pot with ±15V terminal ranges
- One 10A SPDT relay
- One 350mA SPST reed relay
- ±15V and +5V power supply rails
- Remote control via serial terminal and Wi-Fi telnet SCPI commands
- Web interface
- TestController integration
- Powered by a 5V plugpack
- Open-source code (excluding web interface)



☑ B41T.VoltageDC
☑ PlatyKnife.Ain1
☑ PlatyKnife.Ain2
☐ PlatyKnife.Din1
☐ PlatyKnife.Din2
☐ PlatyKnife.Dout1
☐ PlatyKnife.Dout2
☐ PlatyKnife.R1
☐ PlatyKnife.R2
☐ PlatyKnife.DPOT
☐ PlatyKnife.Freq
☑ Math.sVal

First sample to display: ☐ Last sample to display: ☐

☐ hh:mm:ss

Screen 1: here, the DAC output has been fed to both ADC inputs, and we are plotting the desired voltage (mauve) against the actual voltage read by a multimeter (red) and ADC channels one (blue) and two (dark grey) over the range of 0-250mV. The ADC2 plot tracks the external multimeter almost exactly; ADC1 has a slight offset error due to using 5% resistors in the prototype.

is a 10A SPDT type that can switch power supplies and similar. Both have LED indicators.

The maximum recommended voltage across both relays is limited to 50V by safety considerations for breadboard-style operation, rather than the relays themselves. Both can switch in less than 10ms.

The digital inputs and outputs connect to ESP32 3.3V GPIO pins with series resistances to limit current if they are misconnected. The inputs have zener diode protection, will correctly read 5V logic and are tolerant of up to 20V. The inputs and outputs all have LED indicators.

A 256-step 100kΩ digital potentiometer completes the feature set. We have specified a high-voltage type, which allows the pot terminals to be at any voltage within the ±15V analogue supply rails. If you prefer a different resistance (or can't get the 100kΩ type), you can substitute any other MCP45HV value (5kΩ, 10kΩ and 50kΩ).

The project is housed in a UB1 Jiffy box and powered by a 5V plugpack. A readily available switching boost converter module is used to provide ±15V supplies for the op amps and digital pot.

The ±15V and +5V supply rails are available to power external circuitry. The specified boost converter can supply 500mA at +15V and 200mA at –15V.

At idle, the unit draws less than 100mA from the 5V supply and around 200mA with both relays energised and all the LEDs lit. While a 1A plug pack is more than adequate to power the unit itself, we recommend a 1.5A model if you will be powering much in the way of external circuitry.

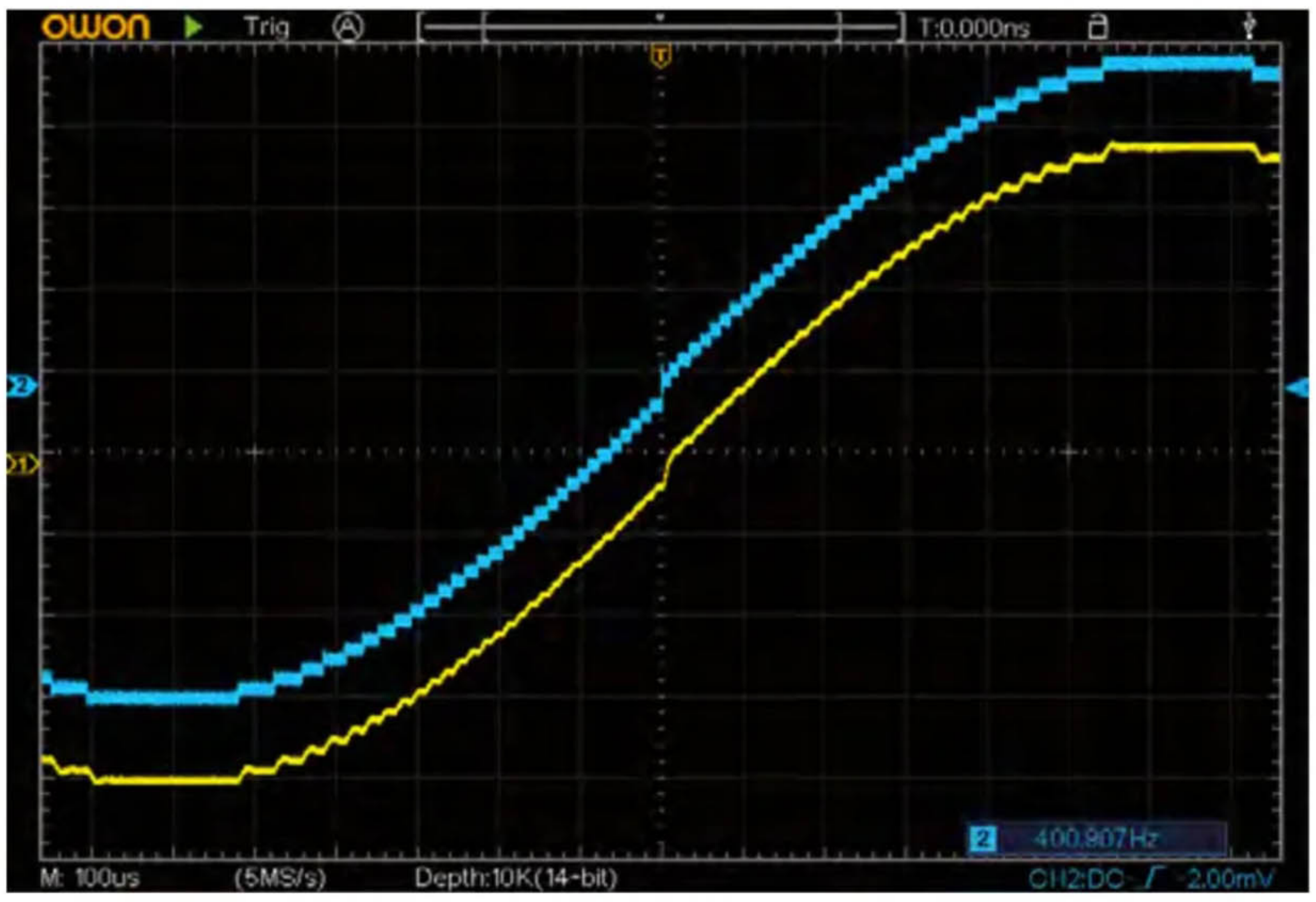
Even with relatively high conversion efficiency, the 5V supply current draw will be around three times that drawn from either the +15V or –15V rails, and more than six times that drawn by a device across those rails.

While the project can be USB-powered for commissioning, the USB cable voltage drop during operation might cause the brownout detector on the ESP32 to trigger, resulting in a potentially endless reboot cycle.

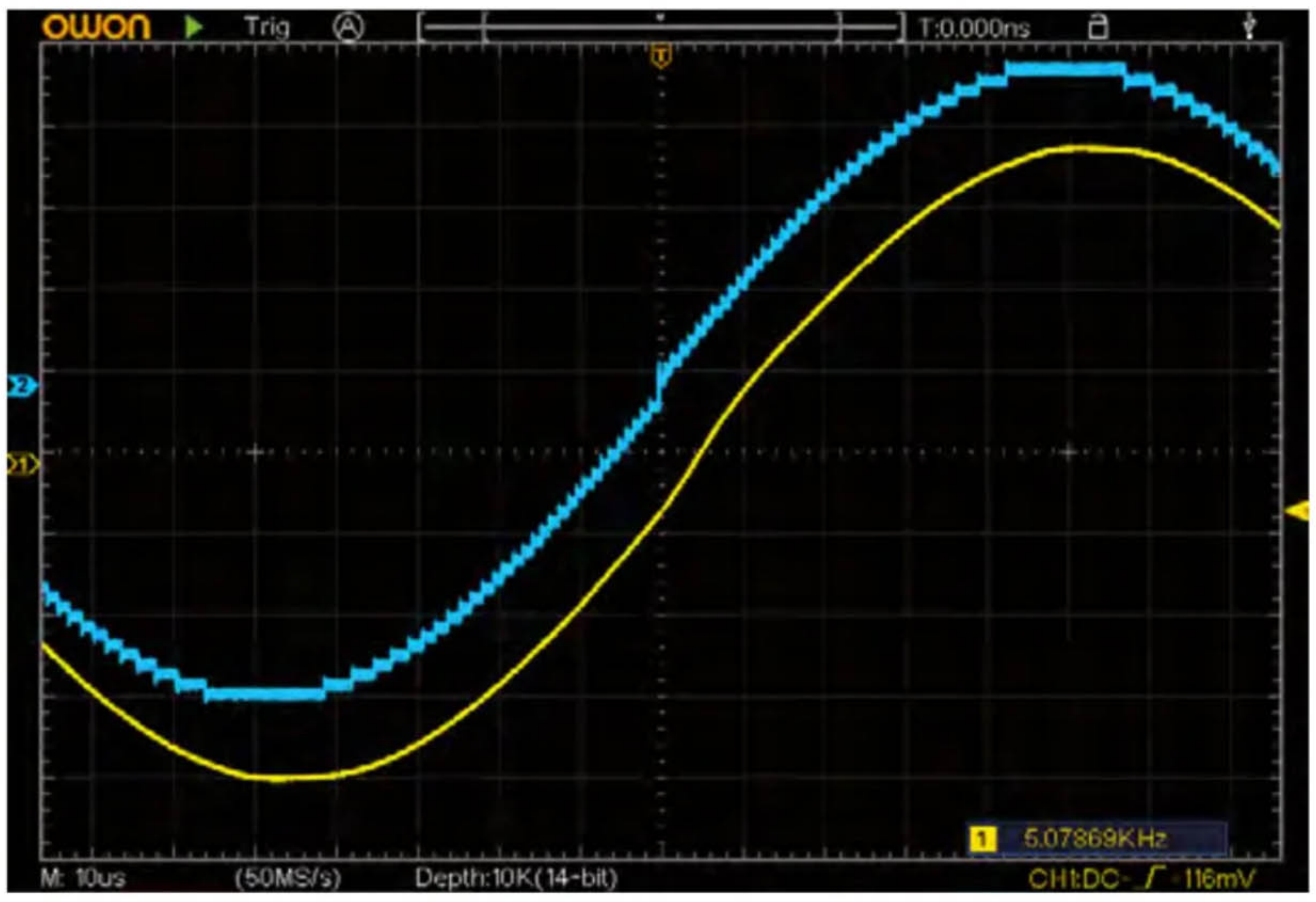
The unit features a flexible suite of remote control functions, which is fortunate as there are no controls on the unit itself! It has been specifically designed to be compatible with TestController, or via its web interface. You can also control it via SCPI text commands from the USB serial monitor in Arduino or via Telnet from a terminal program like PuTTY.

The manual included in the project downloads has full details of the SCPI command set and communication parameters.

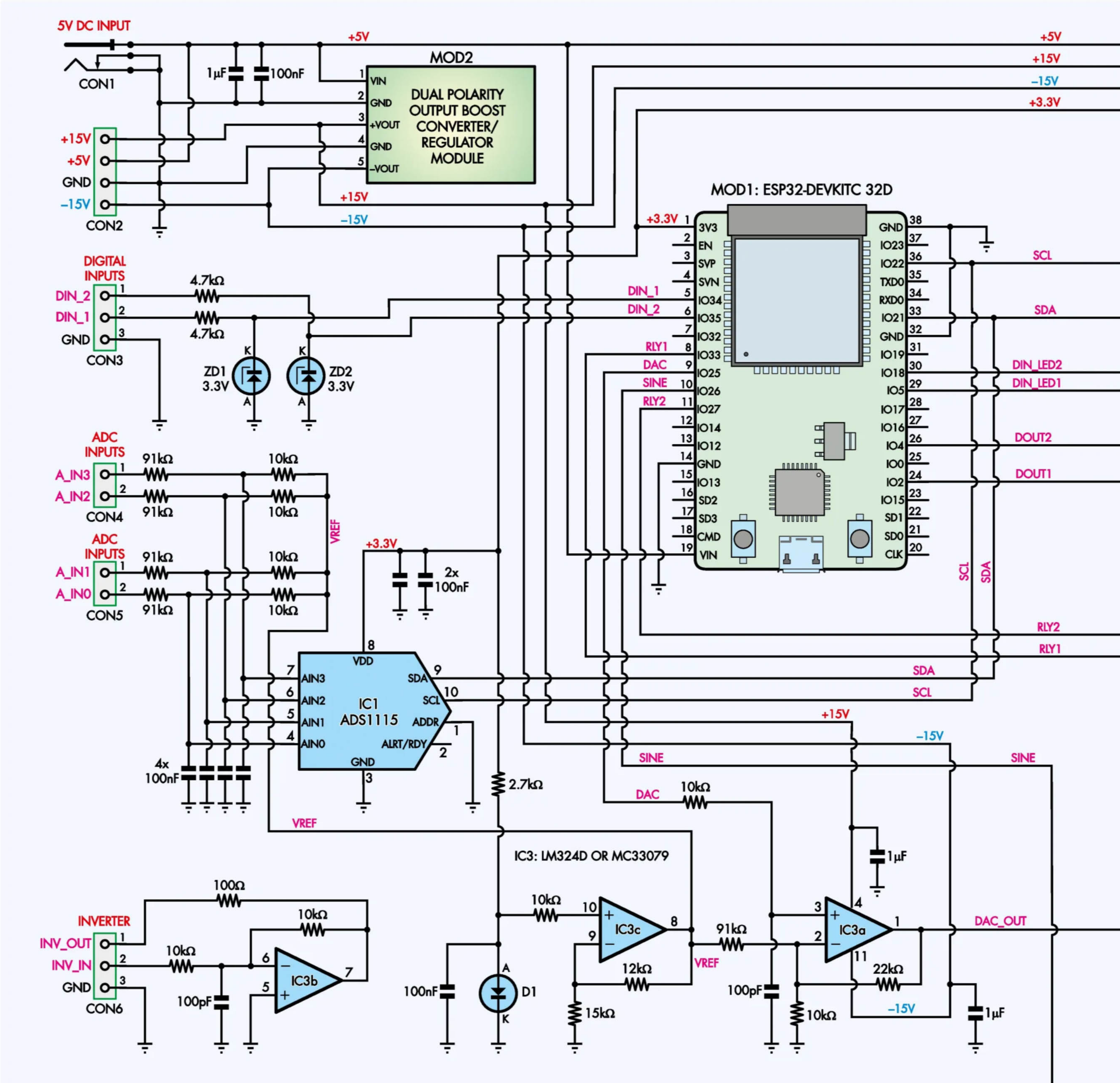
Keeping with our pocket knife theme, we've specified critical resistors as readily available 1% values to provide a full-scale accuracy of a few percent 'out of the box'. With a simple calibration process that only requires a multimeter, you can make the analogue accuracy better than 1%.

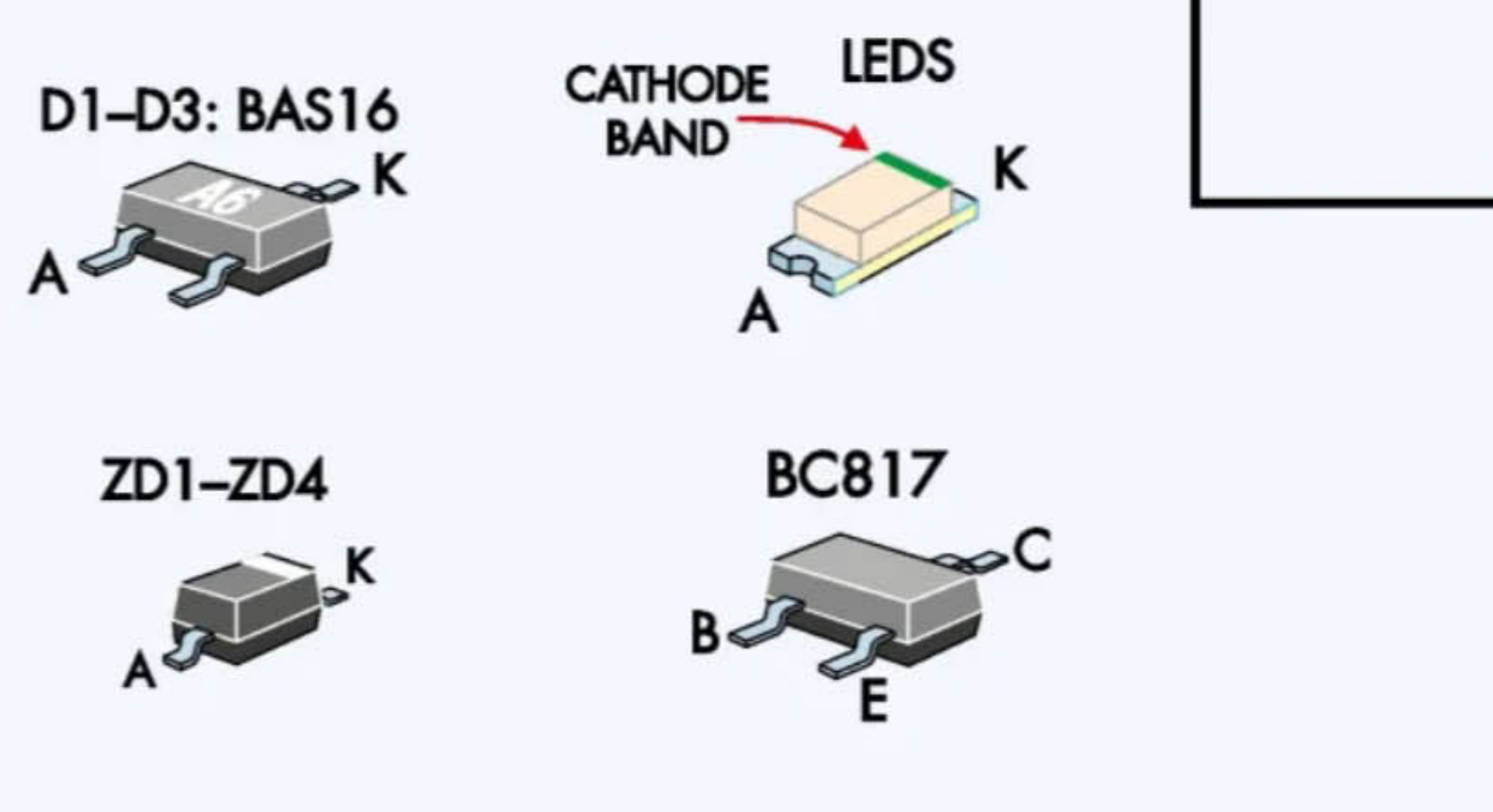


Scope 1: the direct sine output from the DAC at 400Hz (blue trace) on the low-level output range shows some noise and a zero-crossing discontinuity. The filtered output (yellow trace) shows a significant reduction in noise at the cost of a slight overshoot at each step change.



Scope 2: the distortion artefacts from the sinewave output are much less prominent on the filtered output at 5kHz as it spends much less time on each step.


5V DC INPUT
CON1
+5V
+15V
–15V
+3.3V
1µF
100nF
MOD2
VIN
GND
+VOUT
GND
–VOUT
DUAL POLARITY OUTPUT BOOST CONVERTER/ REGULATOR MODULE
+15V
+5V
GND
–15V
CON2
MOD1: ESP32-DEVKITC 32D
3V3
EN
SVP
SVN
IO34
IO35
IO32
IO33
IO25
IO26
IO27
IO14
IO12
GND
IO13
SD2
SD3
CMD
VIN
GND
IO23
IO22
TXD0
RXD0
IO21
GND
IO19
IO18
IO5
IO17
IO16
IO4
IO0
IO2
IO15
SD1
SD0
CLK
DIGITAL INPUTS
DIN_2
DIN_1
GND
CON3
4.7kΩ
ZD1 3.3V
ZD2 3.3V
DIN_1
DIN_2
RLY1
DAC
SINE
RLY2
SCL
SDA
DIN_LED2
DIN_LED1
DOUT2
DOUT1
ADC INPUTS
A_IN3
A_IN2
CON4
A_IN1
A_IN0
CON5
91kΩ
10kΩ
VREF
4x 100nF
2x 100nF
IC1 ADS1115
VDD
AIN3
AIN2
AIN1
AIN0
SDA
SCL
ADDR
ALRT/RDY
GND
2.7kΩ
+15V
–15V
SINE
DAC
IC3: LM324D OR MC33079
INVERTER
INV_OUT
INV_IN
GND
CON6
100Ω
100pF
IC3b
IC3c
IC3a
12kΩ
15kΩ
100nF
D1
22kΩ
DAC_OUT
1µF


D1–D3: BAS16
CATHODE BAND
LEDS
ZD1–ZD4
BC817
A
K
B
C
E

'Swiss Army Knife' Test Bench Multitool

Fig.1: the Swiss Army Knife is based around an ESP32 Wi-Fi microcontroller module. Besides its digital inputs and outputs, its internal DAC at pin 9 (IO25) is used. Because the ESP32 ADC is poor, an external two-channel differential I^{2}C ADC chip (IC1) is used, along with a digital pot IC for that function (IC2) and a quad op amp to buffer and filter the DAC signal plus provide an externally accessible voltage inverter (IC3b).

While this isn't a highly calibrated instrument, it has sufficient flexibility, accuracy and connectivity to make life on the test bench far more productive.

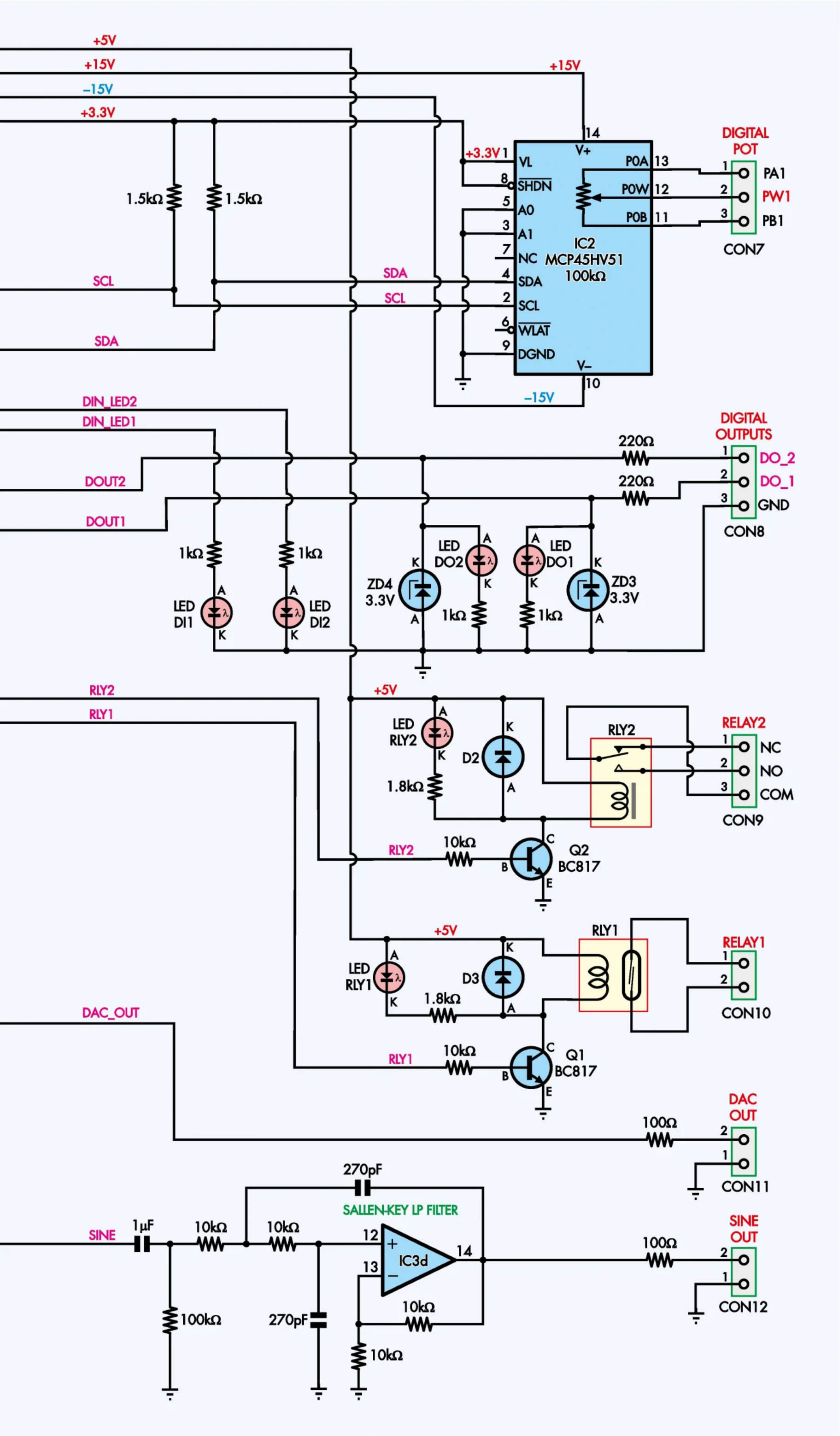
Circuit details

As shown in Fig.1, the heart of the project is an ESP32 Wi-Fi microcontroller module. The ESP32 handles the digital inputs and outputs directly via its GPIO pins, plus it has a DAC and sinewave generator. It also manages Wi-Fi and serial communications.

The nominally 3.3V digital inputs have 4.7kΩ series resistors and 3.3V zener clamping diodes ZD1 and ZD2 to make them reasonably fault tolerant. They draw minimal current from 3.3V logic and around 0.3mA from a 5V source.

The inputs will register 'high' for any voltage above 2.5V at pins 5 and 6 (IO34 and IO35) and are weakly pulled down by 50kΩ resistors within the ESP32.

The input LED indicators are driven by pins 29 and 30 (IO5 and IO18) to avoid loading the digital inputs.


+5V
+15V
–15V
+3.3V
1.5kΩ
1.5kΩ
SCL
SDA
SDA
SCL
+15V
+3.3V
14
V+
VL
SHDN
A0
A1
NC
SDA
SCL
WLAT
DGND
V–
10
–15V
P0A
P0W
P0B
13
12
11
IC2
MCP45HV51
100kΩ
DIGITAL
POT
PA1
PW1
PB1
CON7
DIN_LED2
DIN_LED1
DOUT2
DOUT1
1kΩ
1kΩ
LED
DI1
LED
DI2
A
K
220Ω
220Ω
DIGITAL
OUTPUTS
DO_2
DO_1
GND
CON8
LED
DO2
LED
DO1
ZD4
3.3V
ZD3
3.3V
1kΩ
1kΩ
RLY2
RLY1
+5V
LED
RLY2
D2
1.8kΩ
RLY2
RELAY2
NC
NO
COM
CON9
RLY2
10kΩ
Q2
BC817
B
C
E
+5V
LED
RLY1
D3
1.8kΩ
RLY1
RELAY1
CON10
DAC_OUT
RLY1
10kΩ
Q1
BC817
DAC
OUT
100Ω
CON11
270pF
SALLEN-KEY LP FILTER
SINE
1μF
10kΩ
10kΩ
12
13
14
IC3d
100kΩ
270pF
10kΩ
10kΩ
SINE
OUT
100Ω
CON12

Pins 24 and 26 (IO2 and IO4) drive the digital outputs. When low, they will be below 0.3V, and when high, above 2.7V. 220Ω series resistors limit the output current and, with zener clamping diodes ZD3 and ZD4, provide a measure of protection against misconnection.

Op amp IC3a amplifies the output from the DAC line (pin 9) that's nominally 0-3.3V to 0-10V full scale. The feedback resistor has been chosen to provide a little more than the required three-times gain so that component variations can be readily corrected via calibration.

The 10kΩ resistor and 100pF capacitor form a low-pass filter to reduce the noise from the DAC.

IC3d is an amplifying Sallen-Key low-pass filter for the sinewave output, with a -3dB frequency of around 70kHz. The op amp gain is set to two, as Sallen-Key filters with gains of more than three are unstable.

The MC33079 op amps can drive their outputs within 1.5V of the supply rails and have a 175kΩ input impedance. They can source and sink up to 30mA and feature short-circuit current limiting. 100Ω resistors in series with the outputs provide an extra margin of safety if they are misconnected.

As the op amps use an industry-standard footprint, substitution should be possible if the specified devices aren't available. While the MC33079 is a more modern op amp with better specifications, for most jobs the Swiss Army Knife will be used for, the venerable LM324 will work fine.

While the ESP32 has in-built analogue-to-digital converter (ADC) channels, they are not linear enough for even basic measurements.

Analogue voltages fed in via CON4 and CON5 are measured by a two-channel, 16-bit differential ADS1115 ADC (IC1) which is set to have a 2.048V input range. 91kΩ/10kΩ resistive dividers on the inputs reduce 10V signals to just under 1V, allowing for excess input voltages to be sensed and some component variation to be corrected by calibration.

As it is desired to measure both positive and negative voltages, both divider chains are referenced to the 1.1V bias supply (VREF) rather than ground.

The ADS1115 has inbuilt over-voltage and negative voltage protection for input currents of less than 10mA, which are limited by the upper resistors in the dividers. If the ADS1115 isn't available, an ADS1015 can be substituted with a slight drop in accuracy.

The bias voltage for the ADC is provided by IC3c, which amplifies D1's 0.65V forward voltage to the required 1.1V. This diode is biased with 1mA from the 3.3V rail via a 2.7kΩ current-limiting resistor.

Inverting amplifier IC3b completes the analogue functions. Its gain is set to –1 and input impedance to 10kΩ by the pair of 10kΩ resistors. The 100pF capacitor combines with those resistors to filter noise from the input, with a corner frequency of 160kHz (1 / [2π × 10kΩ × 100pF]).

Digital potentiometer

The terminal voltages of digital pots are generally limited to the device's digital supply rails. The MCP45HV51 (IC2) is a somewhat unusual high-voltage component with an extended analogue-side voltage range. Its ±15V analogue power rails allow the pot terminal voltages to be anywhere within that range. While we chose the 100kΩ model for our prototype, the MCP45HV series also has 5kΩ, 10kΩ and 50kΩ variants, any of which may be substituted without any circuit changes.

Both the ADS1115 ADC and MCP45HV digital pot are controlled over an I^{2}C serial bus by the ESP32. Both devices have their additional address pins tied low.

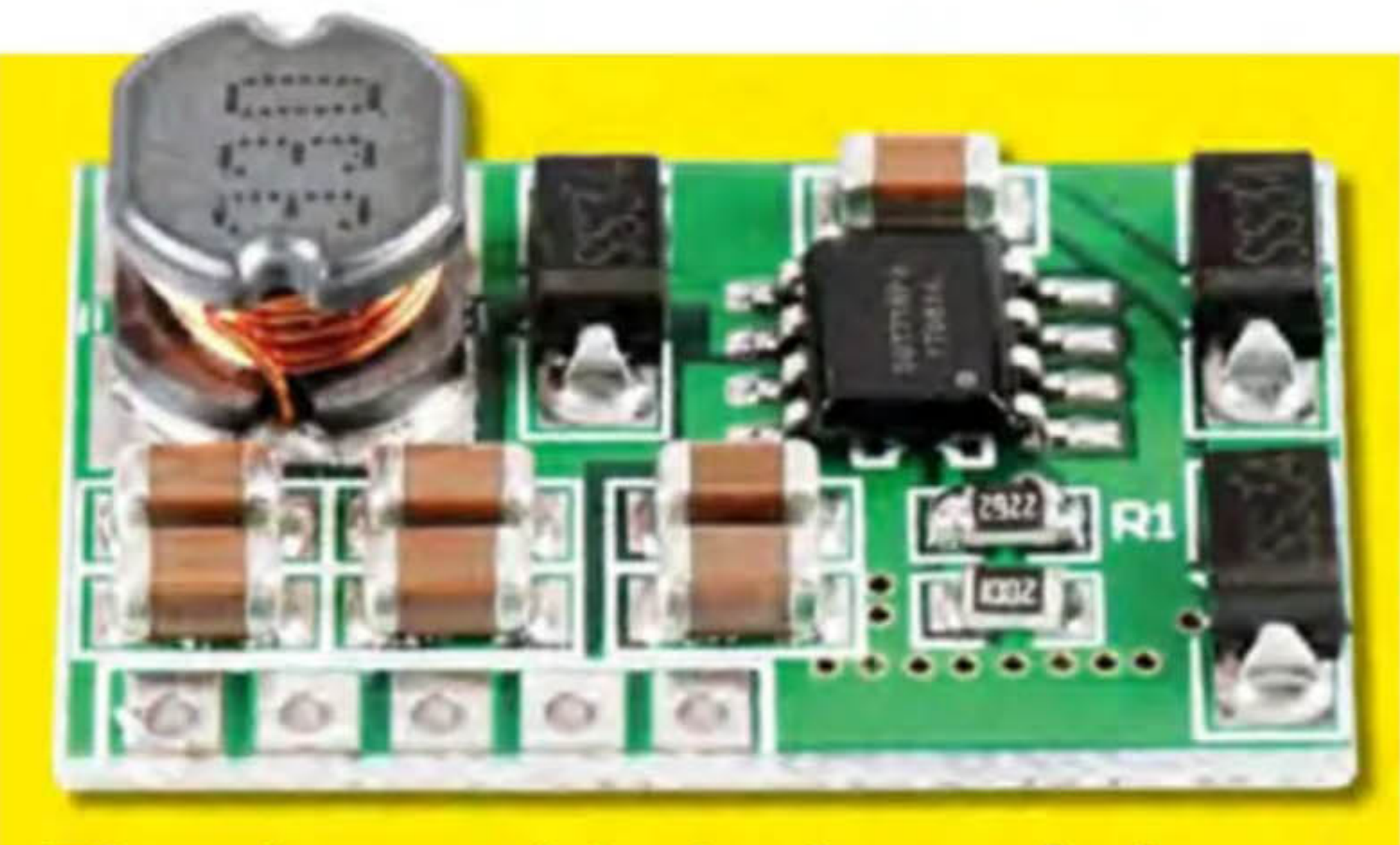
Two relays are provided, driven by NPN transistors Q1 and Q2, with diodes D2 and D3 to quench back-EMF of the coils at switch-off. RLY1 is a 350mA SPST reed relay with a 15mA coil, while RLY2 is a heavy-duty, 10A model with SPDT contacts and a 5V 85mA coil. The indicator LEDs light when a coil is energised.

Power comes from a 5V plugpack and a boost converter module (MOD2) that supplies ±15V. All three supply rails are brought out to a terminal block for breadboard use.

Case preparation

Start by marking out and cutting the holes in the lid as shown in Fig.2. There are just the four corner mounting holes to drill to 3mm, plus the rectangular cut-out to make. You can do that by drilling a series of holes just inside the rectangular outline, cutting between the holes to remove the plastic inside and then filing the edges smooth and to full size.

We're doing this before assembling the PCB because, to assist you in locating the holes, you can place the blank PCB on the underside of the Jiffy box lid with the component side showing. It should sit neatly inside the locating ridges. Mark and drill the four mounting holes, then make the cut-out, which should be 3mm outside the terminal block outlines.



The pins on this dual-supply boost converter (5V to ±15V) match those on the PCB (MOD2). Other 5V to ±15V modules could be used but would need to be wired to the PCB appropriately.

While not necessary, it would be nice to countersink those four mounting holes and use countersunk screws, so they are flush with the lid.

PCB assembly

Given the ongoing global supply shortage of electronic components, some substitution of the active components may be required. Alternatives are noted in the circuit details above and in the parts list.

The 142 × 83.5mm double-sided PCB is coded 04110221, available from the *PE PCB Service*, and the component locations are shown in Figs.3 and 4. Most of the components and the ESP32 are on one side, with just the connectors and LEDs on the other side.

It's best to fit the three SMD ICs first. Locate their pin 1 indicators and line them up with the pin 1 indicators on the PCB or Fig.3. Spread flux paste on the IC pads, then tack one pin of the IC to a corner pad.

Check that the part is flat on the PCB and all the leads line up with the pads, re-check the orientation of pin 1, then tack a diagonally opposite pin. Solder the remaining pins with minimal solder on the iron and clean up any bridges between pins with more flux paste and some solder wick.

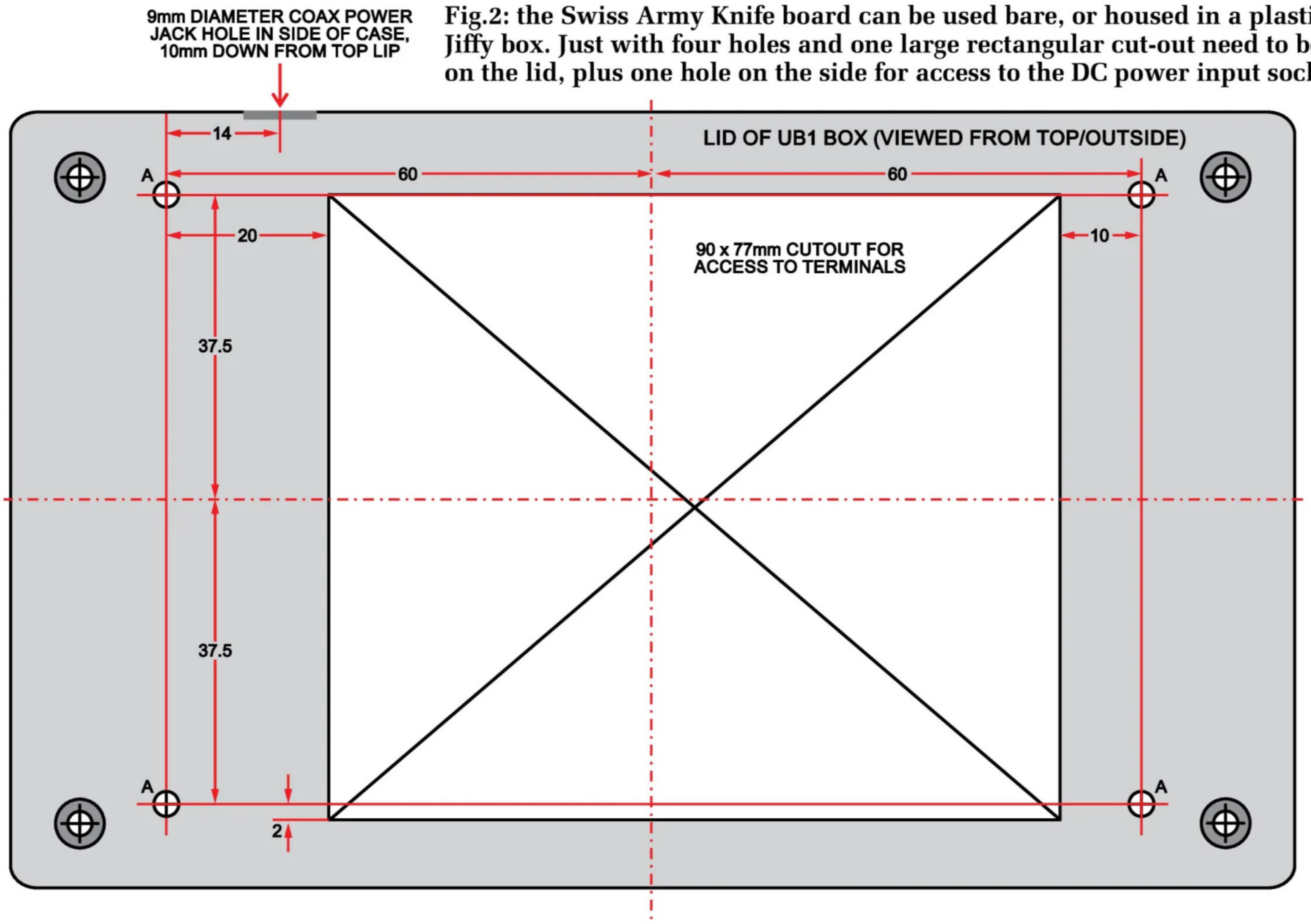
Once you've finished, clean off all the flux residue and scrutinise the pins under magnification to ensure all solder joints have formed properly.

Move on to the four SOT-23 devices and solder them using a similar technique. Note that there are two devices using this package, so don't get them mixed up. Then solder the four zener diodes, ensuring their cathode stripes face as shown.

Follow with the SMD capacitors and resistors; the resistors will be marked with codes indicating their values, but you'll have to refer to the ceramic capacitor packages to see their values (or measure them if unsure).

Now flip the board over and solder the six SMD LEDs using a similar technique. Their cathodes are usually

Fig.2: the Swiss Army Knife board can be used bare, or housed in a plastic UB1 Jiffy box. Just with four holes and one large rectangular cut-out need to be made on the lid, plus one hole on the side for access to the DC power input socket.


9mm DIAMETER COAX POWER JACK HOLE IN SIDE OF CASE, 10mm DOWN FROM TOP LIP
LID OF UB1 BOX (VIEWED FROM TOP/OUTSIDE)
14
60
60
A
A
A
A
20
10
90 x 77mm CUTOUT FOR ACCESS TO TERMINALS
37.5
37.5
2

marked, and they go opposite the + markings in Fig.4 and on the PCB (+ indicates the anodes, not cathodes). You can check their polarity using a DMM set on diode test mode; they should light up with the red lead touching the anode and the black lead touching the cathode.

With all the SMDs on the board, clean off any remaining flux residue before fitting the through-hole parts. We have specified header sockets for the ESP32 and the boost module so you can make those items pluggable. While it might be possible to solder them directly, we don't advise that as it will interfere with the testing and programming sequence.

On the side with most of the components, fit the DC socket (CON1), ESP32 (MOD1), boost module (MOD2) and relays. When fitting the boost module, refer to Fig.3 and the photo above. There is an extra row of pins for the ESP32 on the PCB, as some variants of the ESP32 DevkitC come with narrower spacing. You only need to populate the row that matches your module.

Mount the terminals (CON2-CON12) on the other side of the board. You'll probably want to orient them so that the wire entries face the outside of the board, as that will be the most convenient way to use it.

Final assembly

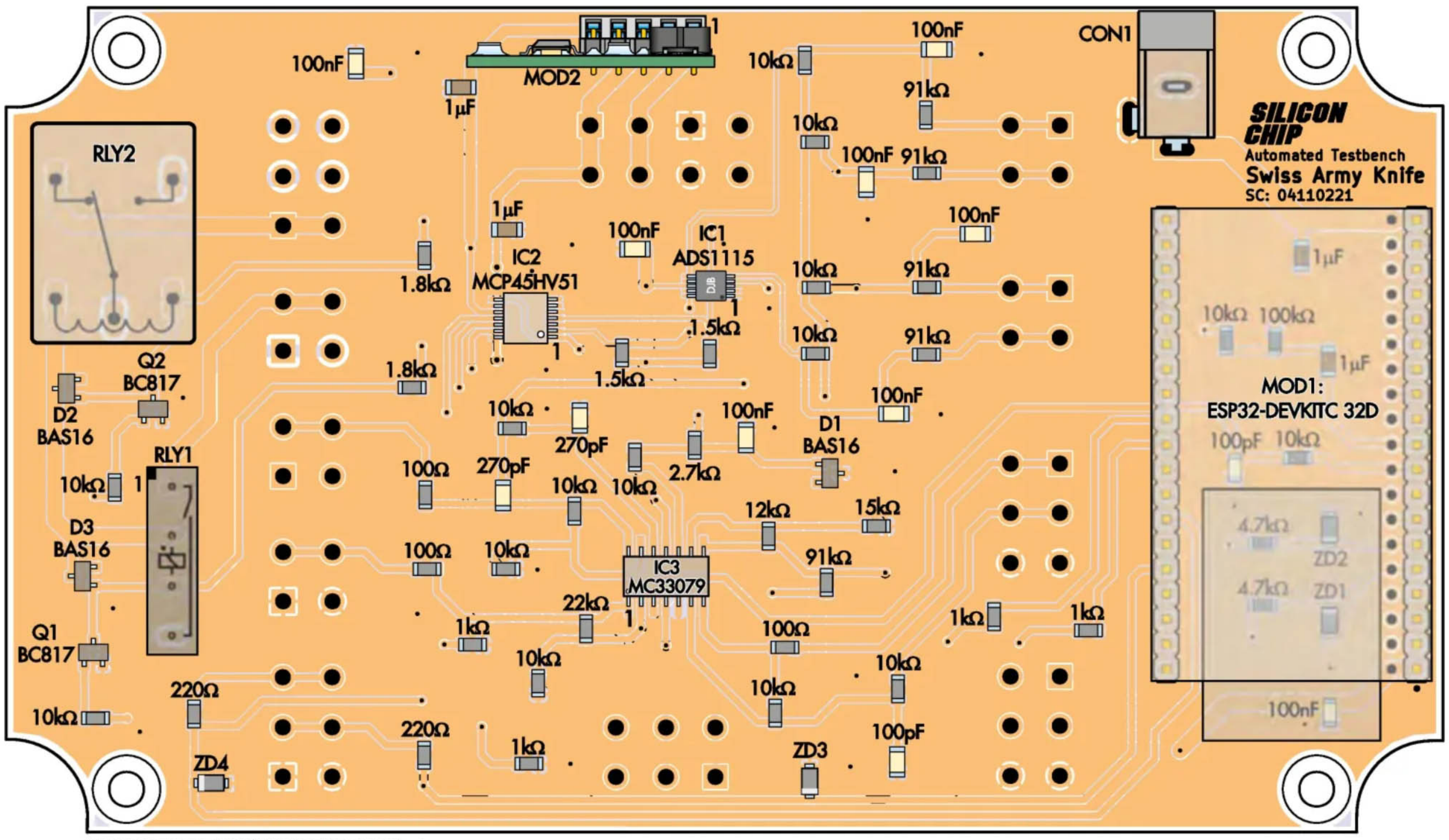
The PCB mounts under the lid of a UB1 jiffy box with a hole cut in its top, exposing the rectangular area shown in Fig.4. It is a tight fit; some trimming of the PCB locating slots on the case's side walls may be required. There is no need for a decal or cover plate as the critical information is silk-screened directly onto the PCB.

Clip or file off any pins protruding more than 1.5mm from the silk-screened side of the board, and mount it on the lid using 2mm spacers (eg, two 1mm-thick washers stacked) to provide clearance for the component pins. Mark and drill the hole in the case for the coaxial power socket, as shown in Fig.2, if you haven't already.

Loading the software

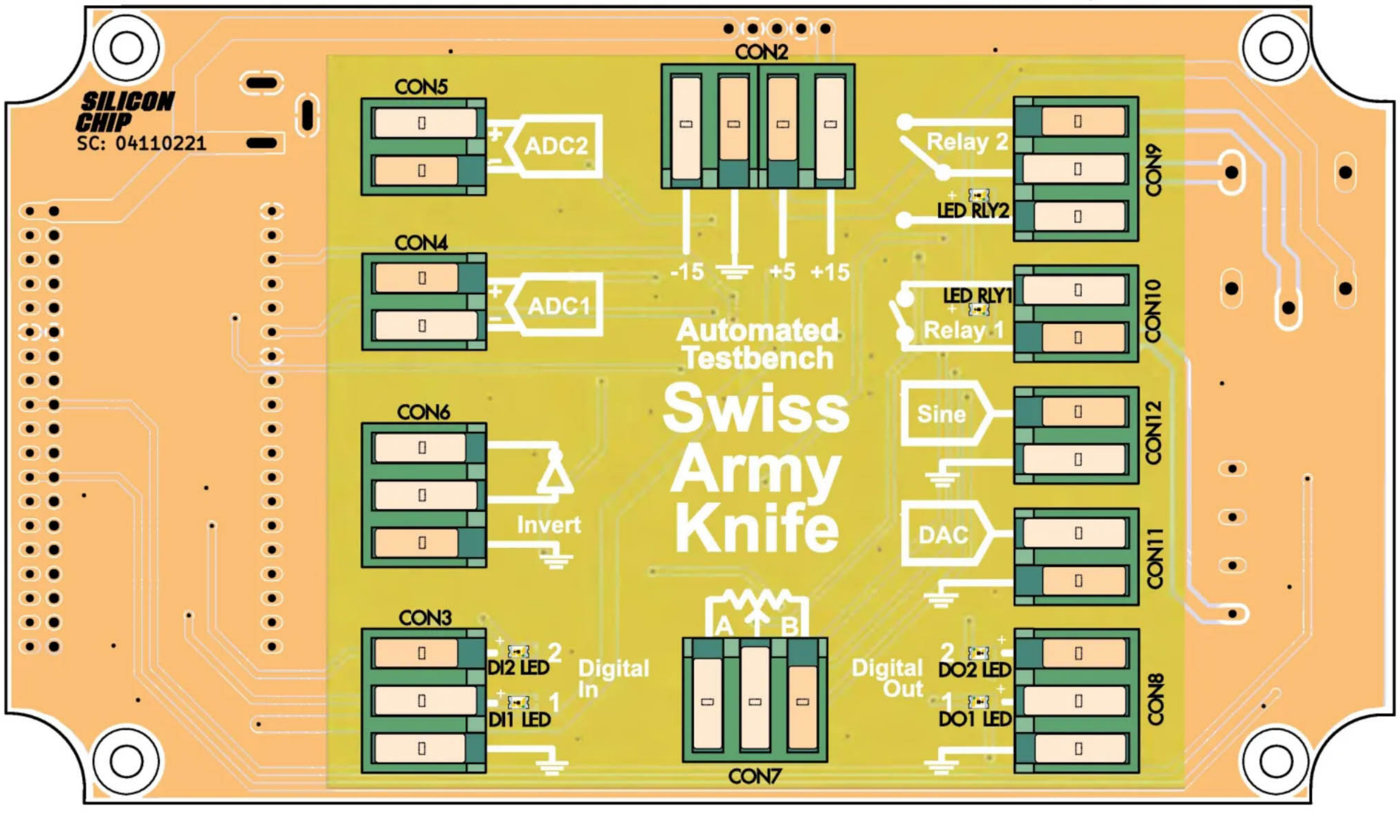
You should now program the ESP32 separated from the PCB. As well as programs being compiled and loaded via an integrated development environment (IDE) such as the Arduino IDE, the ESP32 can load binary files using an over-the-air (OTA) update program. That has the convenience of being able to update its firmware away from your computer.

The first step is to load the OTA program, which also conducts validation of the PCB. Install the Arduino ESP32 board files, following the instructions at **https://bit.ly/pe-apr24-esp1**


RLY2
100nF
MOD2
1µF
10kΩ
91kΩ
CON1
SILICON CHIP
Automated Testbench
Swiss Army Knife
SC: 04110221
IC2
MCP45HV51
IC1
ADS1115
1.8kΩ
1.5kΩ
Q2
BC817
D2
BAS16
RLY1
D3
BAS16
Q1
BC817
270pF
100Ω
2.7kΩ
D1
BAS16
12kΩ
15kΩ
IC3
MC33079
22kΩ
1kΩ
MOD1:
ESP32-DEVKITC 32D
100kΩ
100pF
4.7kΩ
ZD2
ZD1
220Ω
ZD4
ZD3

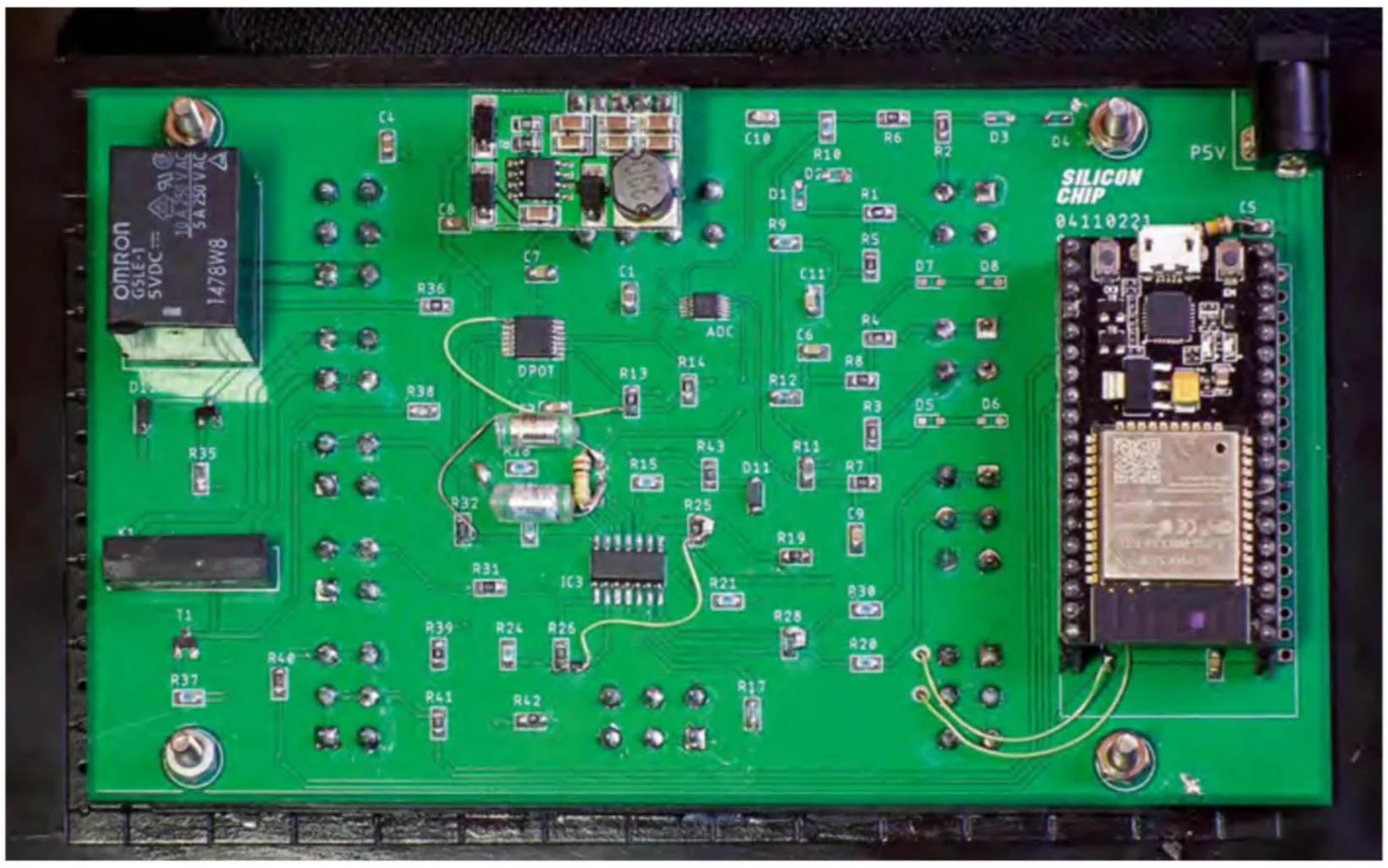
REAR OF PCB (COMPONENT SIDE) ↑

↓ FRONT OF PCB (TERMINAL BLOCK SIDE)


SILICON CHIP
SC: 04110221
CON5
ADC2
CON4
ADC1
CON6
Invert
CON3
DI2 LED
DI1 LED
Digital In
CON2
-15
+5
+15
Automated Testbench
Swiss Army Knife
A
B
CON7
Digital Out
DO2 LED
DO1 LED
Relay 2
LED RLY2
LED RLY1
Relay 1
Sine
DAC
CON9
CON10
CON12
CON11
CON8

Figs.3 and 4: fit the components to the board as shown here, paying particular attention to the orientations of the ICs, LEDs, zener diodes, relay RLY1 and the boost module. Also, don't get the transistors (Q1 and Q2) and small signal diodes (D1 and D2) mixed up. The resistors and capacitors are not polarised; while the resistors will be marked with coded values, the capacitors won't.

While the boost module is shown mounted vertically here, using a straight header, you can mount it horizontally as shown in the photo overleaf.



The underside of the PCB is where most of the components are mounted. This prototype differs from the final version, hence the added wires and components.

Parts List – Test Bench 'Swiss Army Knife'

1 double-sided PCB coded 04110221, 142mm × 83.5mm available from the PE PCB Service
1 UB1 Jiffy box **[Altronics H0201 or H0151, Jaycar HB6011]**
1 laser-cut UB1 Jiffy box lid (**optional**; 3mm acrylic) **[Silicon Chip SC6337]**
1 5V 1A or 1.5A plugpack with 2.1mm inner diameter coaxial plug **[Altronics M8903A, Jaycar MP3144]**
1 Espressif ESP32-DEVKITC-32D (MOD1) **[Silicon Chip SC4447, Altronics Z6385A, Jaycar XC3800]**
1 +5V to ±15V boost regulator module (MOD2) **[Silicon Chip SC6587]**
1 micro-USB cable (to program MOD1)
1 5V SIP reed relay (RLY1) **[Pan Chang SIP-1A05, Littelfuse HE3621A0510, Teledyne SIP-1A05-D]**
1 5V DC coil 10A SPDT relay (RLY2) **[Altronics Z6325, Jaycar XC4419]**
2 19-pin female 2.54mm headers (for MOD1)
1 5-pin female 2.54mm header (for MOD2) (can be cut from longer header)
1 2.1mm inner diameter PCB-mount DC barrel socket (CON1) **[Altronics P0620, Jaycar PS0519]**
7 2-pole, 5mm pitch 'Euro' type spring terminal blocks (CON2, CON4, CON5, CON10-CON12) **[Altronics P2068, Jaycar HM3140, DECA MX722-500M or Eaton EM278502]**
5 3-pole, 5mm pitch 'Euro' type spring- or cam-operated terminal blocks (CON3, CON6-CON9) **[Altronics P2070, Jaycar HM3142, DECA MX732-500M or Eaton EM278503]**
4 M3 × 12mm countersunk machine screws and hex nuts
8 M3 x 1mm Nylon washers

Semiconductors
1 ADS1115IDGST or ADS1115IDGSR ADC, MSOP-10 (IC1)
1 MCP45HV51-x0xE/ST 8-bit I^2C digital potentiometer, TSSOP-14 (IC2) (x0x = 502 [5kΩ], 103 [10kΩ], 503 [50kΩ] or 104 [100kΩ])
1 LM324D or MC33079 quad op amp, SOIC-14 (IC3) **[Altronics Y2523, Jaycar ZL3342]**
2 BC817 or BC846-BC850 SMD NPN transistors, SOT-23 (Q1, Q2) **[Altronics Y1312, Jaycar ZT2118]**
6 SMD LEDs, M2012/0805 or gull-wing **[Altronics Y1107, Jaycar ZD2000]**
4 3.3V 1/2W+ zener diodes, DO-214AC or DO-213AA/SOD-80/MiniMELF (ZD1-ZD4) **[eg, BZG05C3V3 or MLL5226B]**
3 BAS16, BAV99 or similar signal diode, SOT-23 (D1-D3) **[Altronics Y0089]**

Capacitors *(all 50V SMD ceramic M2012/0805 size)*
4 1µF X7R
8 100nF X7R
2 270pF NP0
2 100pF NP0

Resistors *(all 1% SMD metal film, M2012/0805 size)*
1 100kΩ
5 91kΩ
1 22kΩ
1 15kΩ
1 12kΩ
15 10kΩ
2 4.7kΩ
1 2.7kΩ
2 1.8kΩ
2 1.5kΩ
4 1kΩ
2 220Ω
3 100Ω

Next, install the ESP32 exception decoder and file uploader plug-in releases: **https://bit.ly/pe-apr24-esp2**

Select 'ESP32 Dev Module' as the board in the Tools menu of the Arduino IDE and edit the **OTA-Test.ino** file from the project download package to include your Wi-Fi credentials – file available from the April 2024 page of the *PE* website: **https://bit.ly/pe-downloads**

Compile and run the program; the Serial Monitor will display the IP Address of the ESP32. You should get an output similar to Screen 2 with the Arduino Serial Monitor baud rate set to 115,200. As expected, the program has failed to find the ADC and digital pot. If you miss the messages on the Serial Monitor, simply push the boot (EN) button on the ESP32 module, and it will restart.

Power down the ESP32 and plug it into the PCB sockets with the USB socket near the power input barrel socket, leaving off the boost module for now. Re-connect its USB cable to the computer. The two I^2C devices should now show as available. All six LEDs and the two relays should turn on and off at two-second intervals.

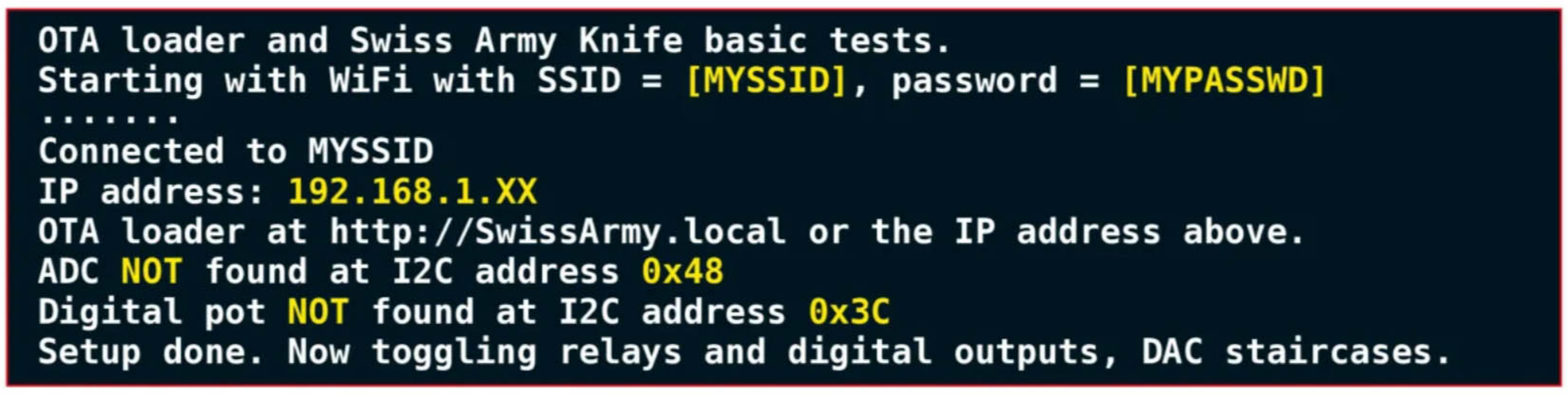
Now connect the boost converter (with power briefly removed) and check the ±15V rails while still operating on USB power. The DAC output should vary slowly between 0 and 10V at the terminal block. The sinewave output should be a series of pulses at the terminal block, as its buffer is AC-coupled, and we're feeding it a staircase signal.

Connect the DAC signal to the inverter input and check that the inverter's output varies inversely with its input voltage. You can fully test the digital pot and ADC once the main program is loaded. For now, we have confirmed that they are responding to I^2C messages.

In the Data folder that is associated with the OTA-Test program, edit the **profile.json** file, find the section that looks like the following and replace the placeholder 'ssid' and 'pass' values with those for your Wi-Fi network:

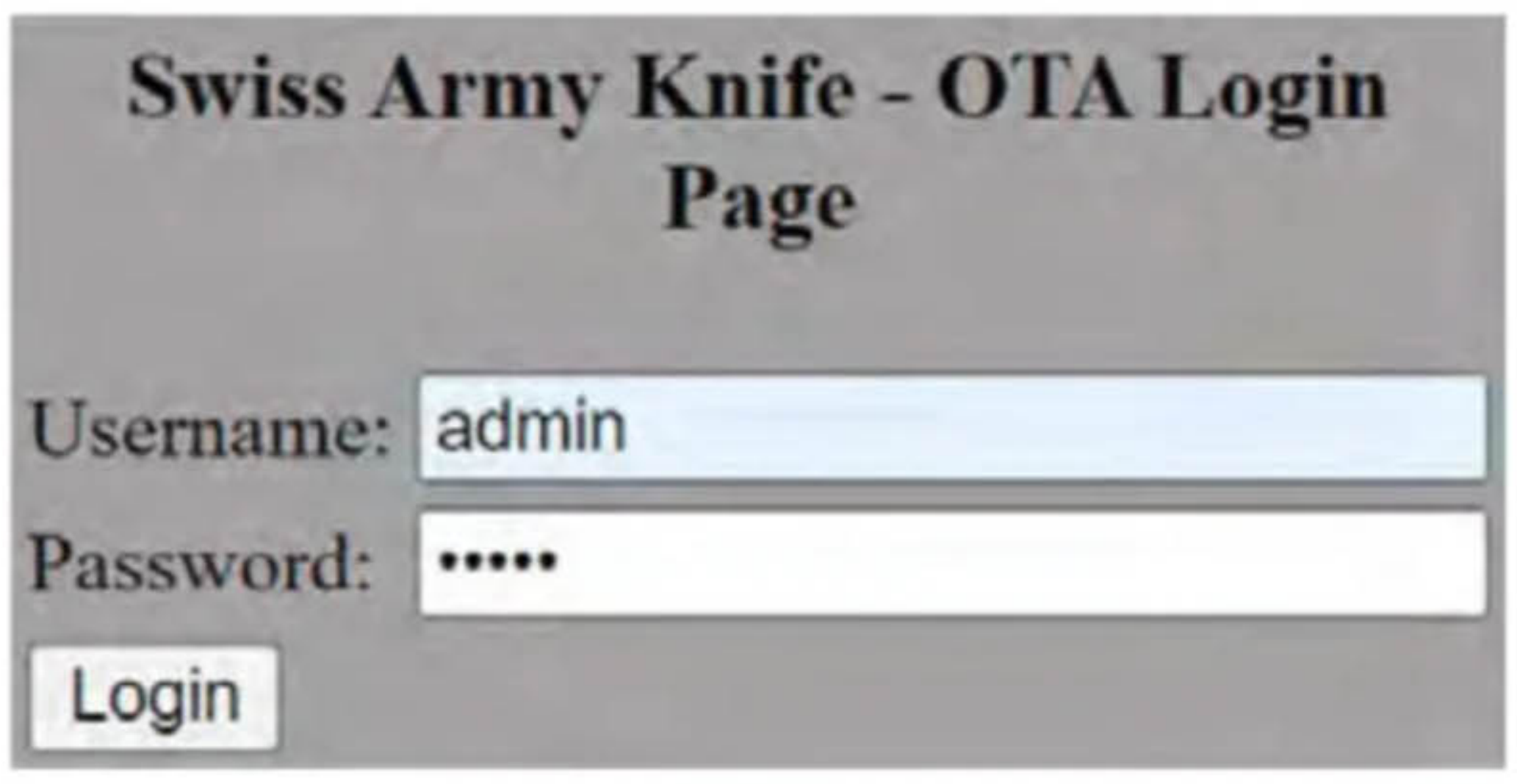
```
{
  “ssid” : “your SSID”,
  “pass” : “WiFi password”,
  “hostname” : “SwissArmy”
}
```

Next, close the Serial Monitor window. In the Arduino Tools menu, click 'ESP32 Sketch Data Upload' to copy the files in the Data folder to the ESP32's local file system. The rest of the files in this folder are needed for the web interface. This uploaded file system will remain intact when new programs are uploaded.



```
OTA loader and Swiss Army Knife basic tests.
Starting with WiFi with SSID = [MYSSID], password = [MYPASSWD]
.......
Connected to MYSSID
IP address: 192.168.1.XX
OTA loader at http://SwissArmy.local or the IP address above.
ADC NOT found at I2C address 0x48
Digital pot NOT found at I2C address 0x3C
Setup done. Now toggling relays and digital outputs, DAC staircases.
```

Screen 2: the expected output of the OTA-Test program on the serial monitor, before the ESP32 is plugged into the main PCB.



Swiss Army Knife - OTA Login Page

Username: admin

Password: •••••

Screen 3: the Over The Air (OTA) login page displays when first accessing the ESP32 via a browser.

Open up a web page using the IP address or URL indicated by the Serial Monitor. On the OTA-Test program's web interface, log in using 'admin' and 'admin' as the credentials (see Screen 3).

After you have logged in, select the downloaded project BIN file with the 'Choose file' button, and then press the Update button. The web page will track the upload progress, and after a short delay, the ESP32 will reboot.

Re-open the Arduino Serial Monitor, and start-up commands should be displayed, ending with an 'SCPI Command?' prompt. If you type '*IDN?' (without quotes) into the command field on the Serial Monitor and click Send, the software should respond with something like 'Platy,SwissArmy,00,v0.1'.

The unit can now be sealed up in the Jiffy box, powered via the plugpack and remotely controlled via the web interface.

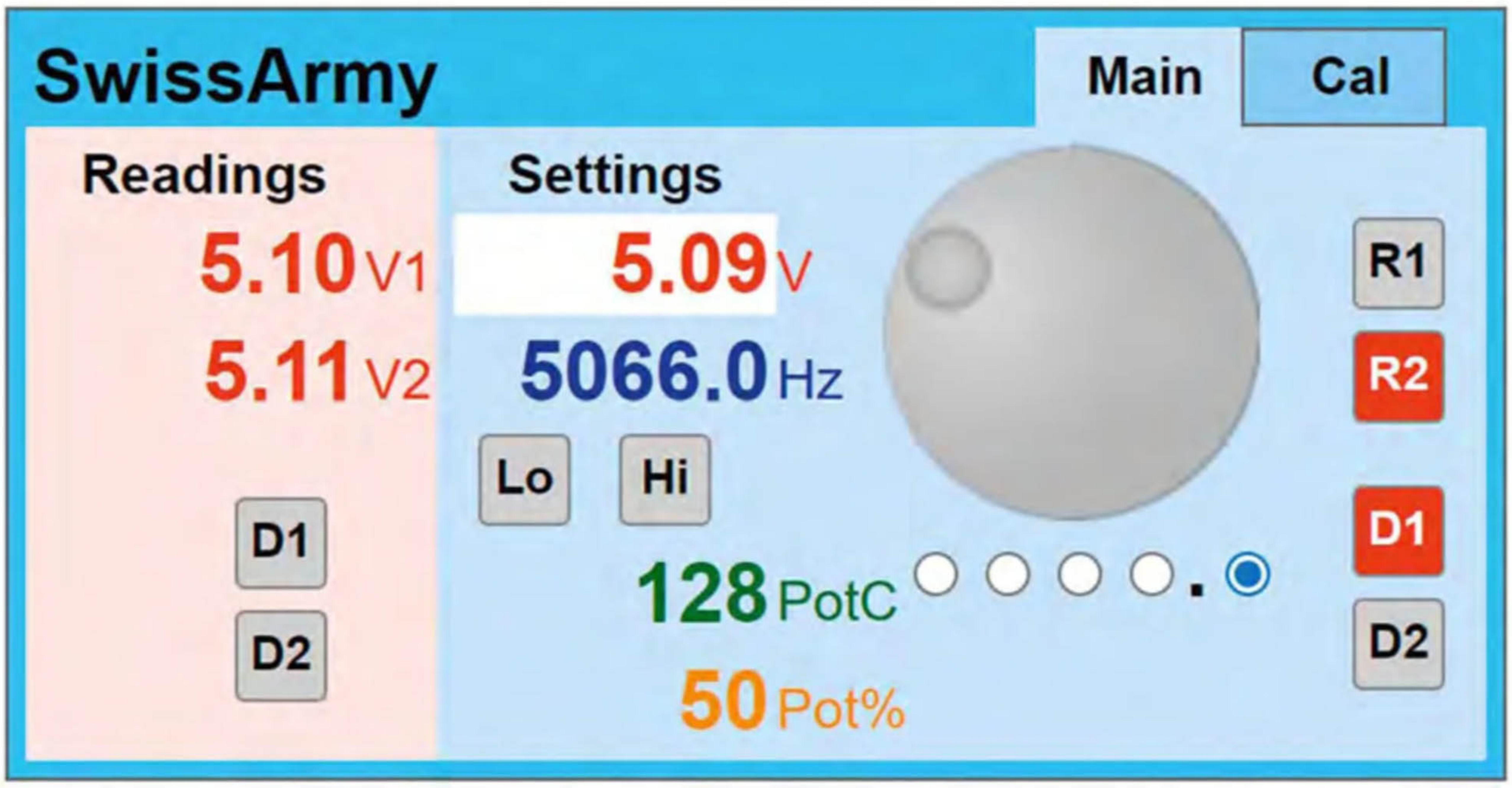
If you intend to use a USB connection from this point on, then we strongly recommend that a USB isolator be used. This will help to avoid damage to the ESP32 in the event of a misconnection.

Changing the Wi-Fi credentials

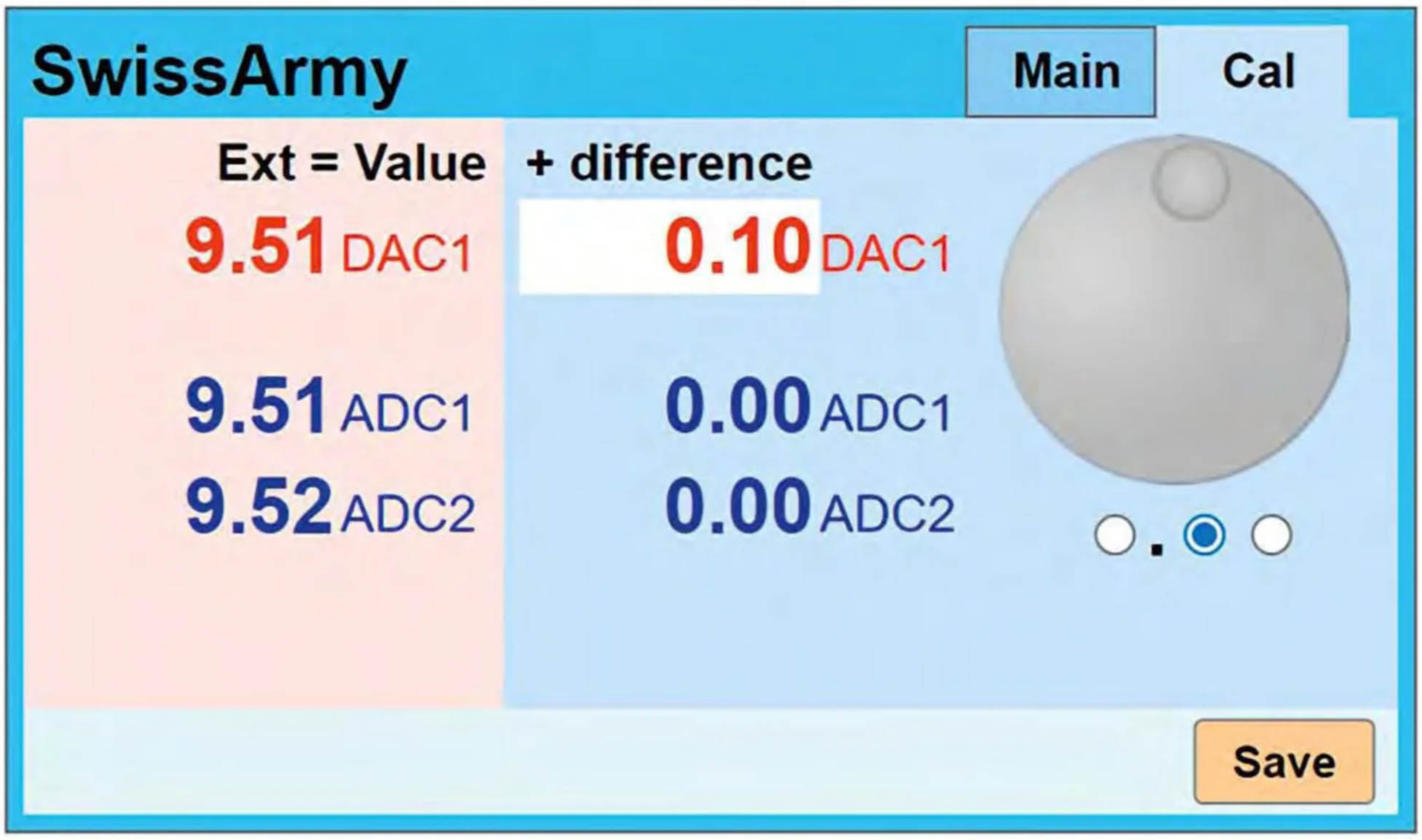
If you have difficulty connecting to your local Wi-Fi or need to change the settings, you can issue the following commands from a terminal program or the Arduino Serial Monitor:

```
:SYST:SSID your-WiFi-SSID-
  without-quote-marks
:SYST:PASS your-WiFi-Password-
  without-quote-marks
```

You can also change the Wi-Fi credentials by editing the **profile.json** file on your computer and uploading it again, using the instructions above. You only need to open the OTA-Test

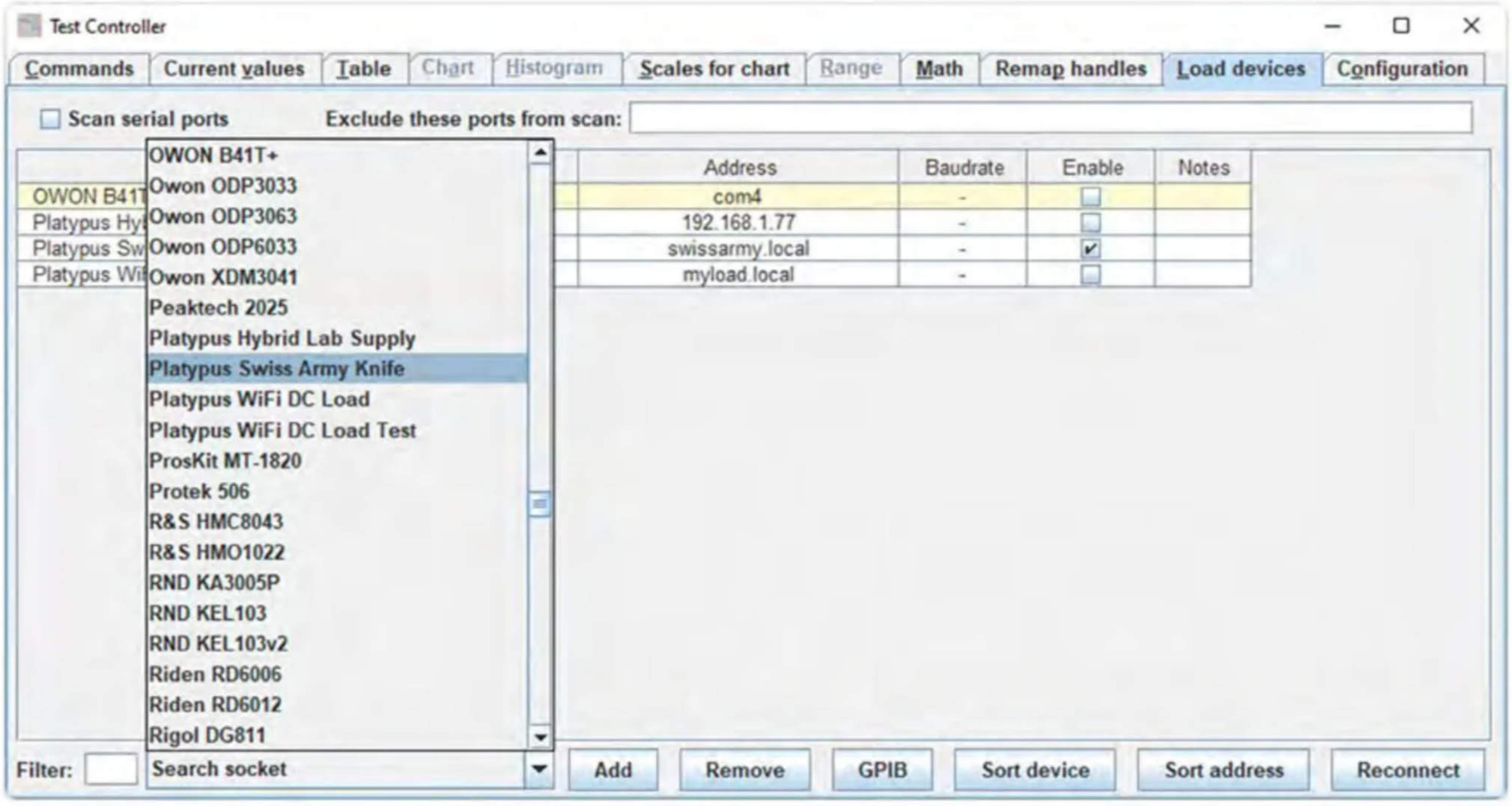


Screen 4: the Swiss Army Knife web interface main page. ADC1 and ADC2 are reading 5.10V and 5.11V respectively, while digital inputs D1 and D2 are both low. On the Settings panel, relay RLY2 is on, and digital output D1 is high. The digital pot is set at 128 steps (50%). The sinewave is currently being adjusted (setting highlighted) to 5.09V; turning the dial will result in 0.1V steps (radio buttons under the dial).



Screen 5: the calibration page. If the external multimeter reads 9.61V, DAC1's output voltage reading would need to be boosted by 0.1V. Changes are not stored until the Save button is clicked but calibration values are saved between sessions.

***The source code and other software files are available from GitHub at:* https://bit.ly/pe-apr24-sak**


Test Controller
Commands
Current values
Table
Chart
Histogram
Scales for chart
Range
Math
Remap handles
Load devices
Configuration
Add
Remove
GPIB
Sort device
Sort address
Reconnect
OWON B41T+
Owon ODP3033
Owon ODP3063
Owon ODP6033
Owon XDM3041
Peaktech 2025
Platypus Hybrid Lab Supply
Platypus Swiss Army Knife
Platypus WiFi DC Load
Platypus WiFi DC Load Test
ProsKit MT-1820
Protek 506
R&S HMC8043
R&S HMO1022
RND KA3005P
RND KEL103
RND KEL103v2
Riden RD6006
Riden RD6012
Rigol DG811

☐ Scan serial ports   Exclude these ports from scan:

	Address	Baudrate	Enable	Notes
OWON B41T	com4	-	☐	
Platypus Hy	192.168.1.77	-	☐	
Platypus Sw	swissarmy.local	-	☑	
Platypus Wi	myload.local	-	☐	

Filter: Search socket

Screen 6: adding the Swiss Army Knife via TestController's 'Load devices' screen. The option won't be available until you've installed the device definition file and restarted TestController.

program and re-upload the sketch data. The OTA-Test program does not need to be compiled or uploaded, but the unit will need to be re-calibrated after the profile upload.

Remote control and calibration

The unit has been primarily designed to work with the open source software TestController: **https://bit.ly/pe-apr24-test** or via its web interface. SCPI commands can also be issued via an isolated USB serial connection or over Wi-Fi, using a terminal program (PuTTy or TeraTerm).

TestController uses SCPI commands to control all functions besides calibration and communication settings. Further details of the remote control modes and SCPI commands are available in the manual included in the download for this project available from the April 2024 page of the *PE* website: **https://bit.ly/pe-downloads**

The web interface can control all the outputs and display all the input readings on its Main tab (Screen 4). It also offers calibration functions on its Cal tab (Screen 5).

It's best if only one of the remote control options is active at any time, as settings made on one interface may not seamlessly update on all the others.

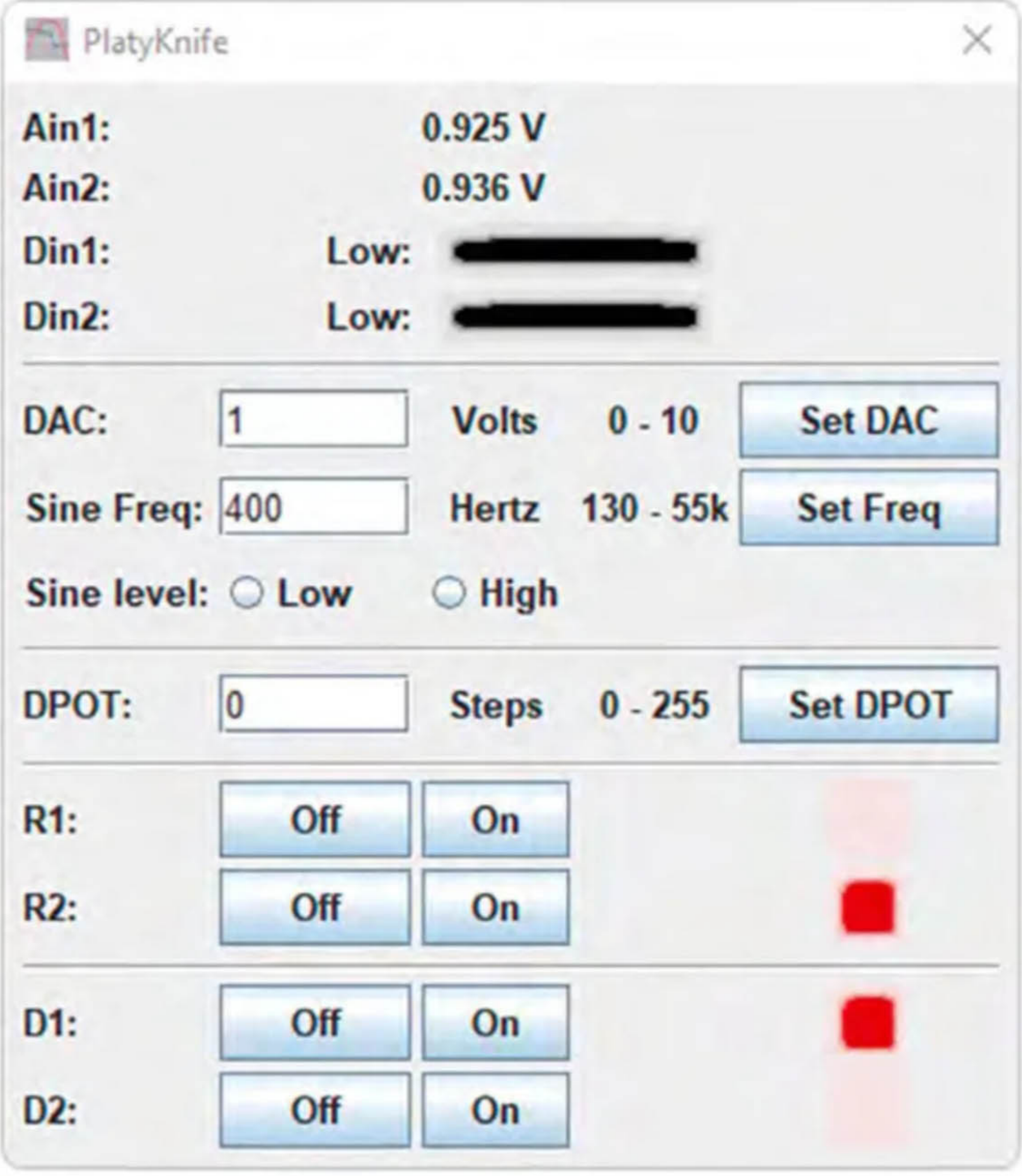
Web interface

The Main tab of the web interface is accessible via **http://swissarmy.local** and has the input readings on the left and settings on the right.

To set a numeric value, click on the setting to be changed and wind the knob. The radio buttons under the knob determine the size of the increment, from 0.1 to 100 units.

Under the sinewave generator frequency setting there are buttons to select the low and high output ranges.

The digital pot has two linked scales, one in counts (0 to 255) and the other in percent of rotation. Either may be used, and the other will change synchronously. The relay and digital output buttons are on the far right.


PlatyKnife
Set DAC
Set Freq
Set DPOT
Off
On

Ain1: 0.925 V
Ain2: 0.936 V
Din1: Low:
Din2: Low:
DAC: 1 Volts 0 - 10
Sine Freq: 400 Hertz 130 - 55k
Sine level: ○ Low ○ High
DPOT: 0 Steps 0 - 255
R1: Off On
R2: Off On
D1: Off On
D2: Off On

Screen 7: the TestController Setup pop-up window shows the readings and allows most functions to be controlled. Input values are updated every second.

Calibration

The analogue inputs and outputs can be calibrated using a multimeter on the Cal tab. Connect the analogue output to both analogue inputs, set the DAC value to around 9.5V on the Main tab then move to the Cal tab.

Measure the analogue input voltage with your multimeter and set the difference between the external multimeter's reading and the analogue input in the 'difference' column for each input (positive if the multimeter reading is higher than shown). Once that is done, set the difference value for the DAC, then click the Save button.

DAC calibration is somewhat less accurate than for the ADCs, given that it only has 256 steps to cover the entire 10V range.

You don't need to calibrate all the inputs and outputs at once as the calibration for any input or output, where the difference value is zero, will remain unchanged when Save is clicked.

TestController integration

The TestController interface can control all functions other than calibration and communication parameters. The device definition file included in the downloads (**SwissArmyKnife.txt**) needs to be loaded into the Devices folder wherever you have installed TestController; the default location is **C:\TestController\Devices**

Restart TestController and add the device on the Load devices tab in TestController (Screen 6), using the address 'swissarmy.local' rather than its IP number, which could change if the unit hasn't been used for some time. Then click the Reconnect button.

On the TestController command screen, click the Setup button, and the pop-up window in Screen 7 should appear. The input values displayed at the top of the window will update once a second, and you can set all output values in the lower sections.

Conclusion

This is a relatively simple project, but it can improve both the productivity and flexibility of your test bench.

Reproduced by arrangement with
SILICON CHIP magazine 2024.
www.siliconchip.com.au

Using the Swiss Army Knife to test itself

The performance graphs in Screen 1 and Screen 8 were created by connecting the analogue output to an analogue input on the unit, then using TestController to control the analogue output. The values were logged by TestController, along with voltage measurements from a Bluetooth-connected multimeter.

TestController was used to create the charts. The results could also have been exported to Excel for analysis. While I wrote a script (shown adjacent) to do this, TestController has a built-in step generator function that would have worked equally well.

I ran the script several times with different parameters. The first iteration tested the basic linearity of the device before calibration, using 0.25V steps to ramp the control value (Math.sVal) from 0 to 10V.

The analogue input (blue) line in Screen 8 is almost hidden behind the multimeter results (red), indicating excellent linearity. The analogue output (grey) had not been calibrated before the test run and shows a full-scale error of around 300mV.

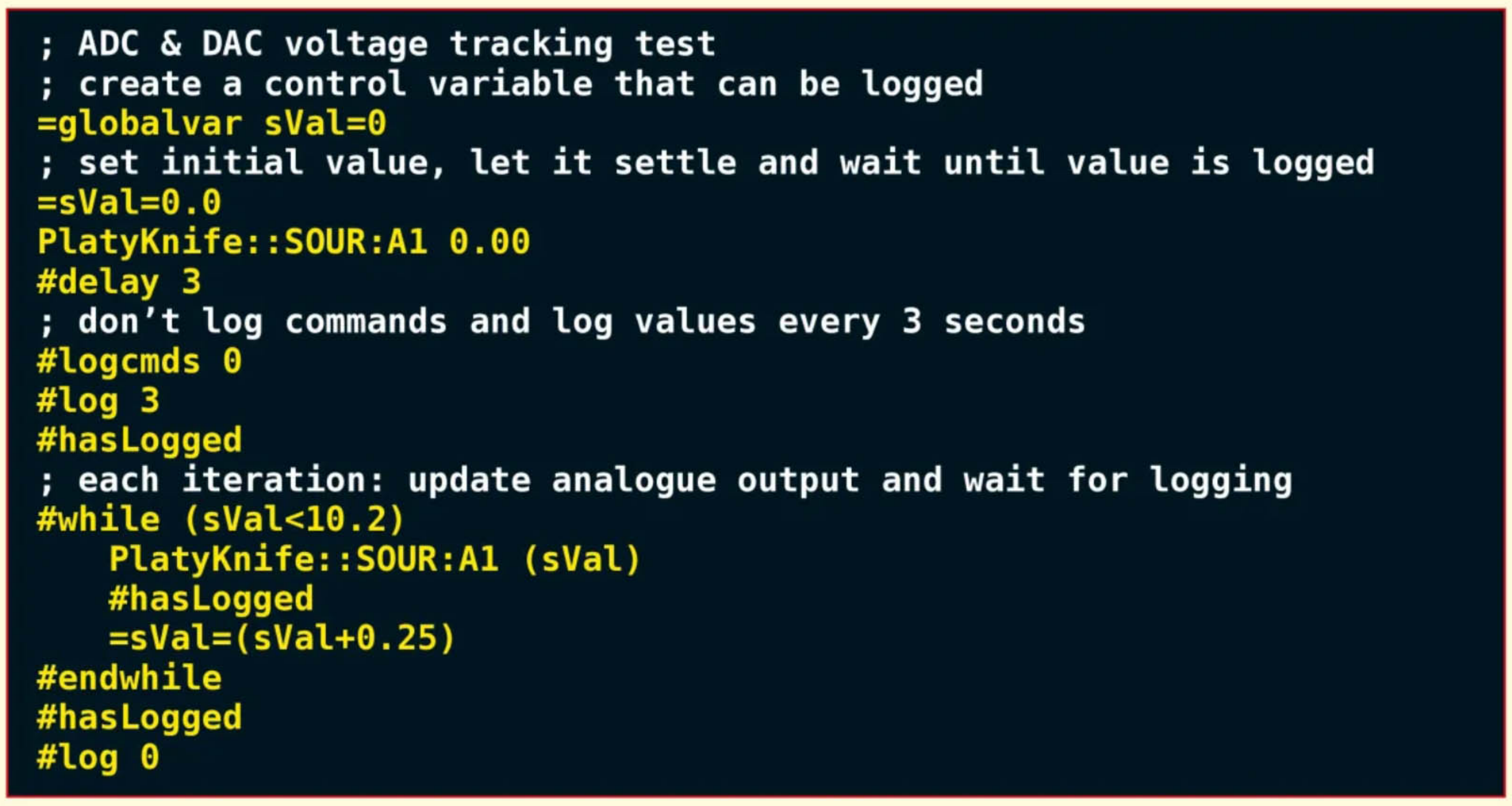
The second test (Screen 1), using increments of 10mV, tested behaviour close to 0V and how the floating-point control value mapped to the 256-step DAC output voltage. As the analogue output has a step size of 40mV, the output voltage stays the same for four 10mV control variable increments, allowing time for each output level to be sampled four times pre-step.

The ADC1 input has a negative offset of -10mV. This was traced to a mismatch between the divider resistors R2 and R4, as 5% 10kΩ resistors were used in the prototype. The second analogue input (dark grey trace) shows almost no offset voltage and tracks the multimeter reading accurately across the entire range.

The code averages 16 samples per reading to reduce the variation between readings. The ADS1115 is capable of 860 samples per second. Over the two ADC channels, averaging sixteen samples gives 25 readings per second, more than fast enough for our purposes.

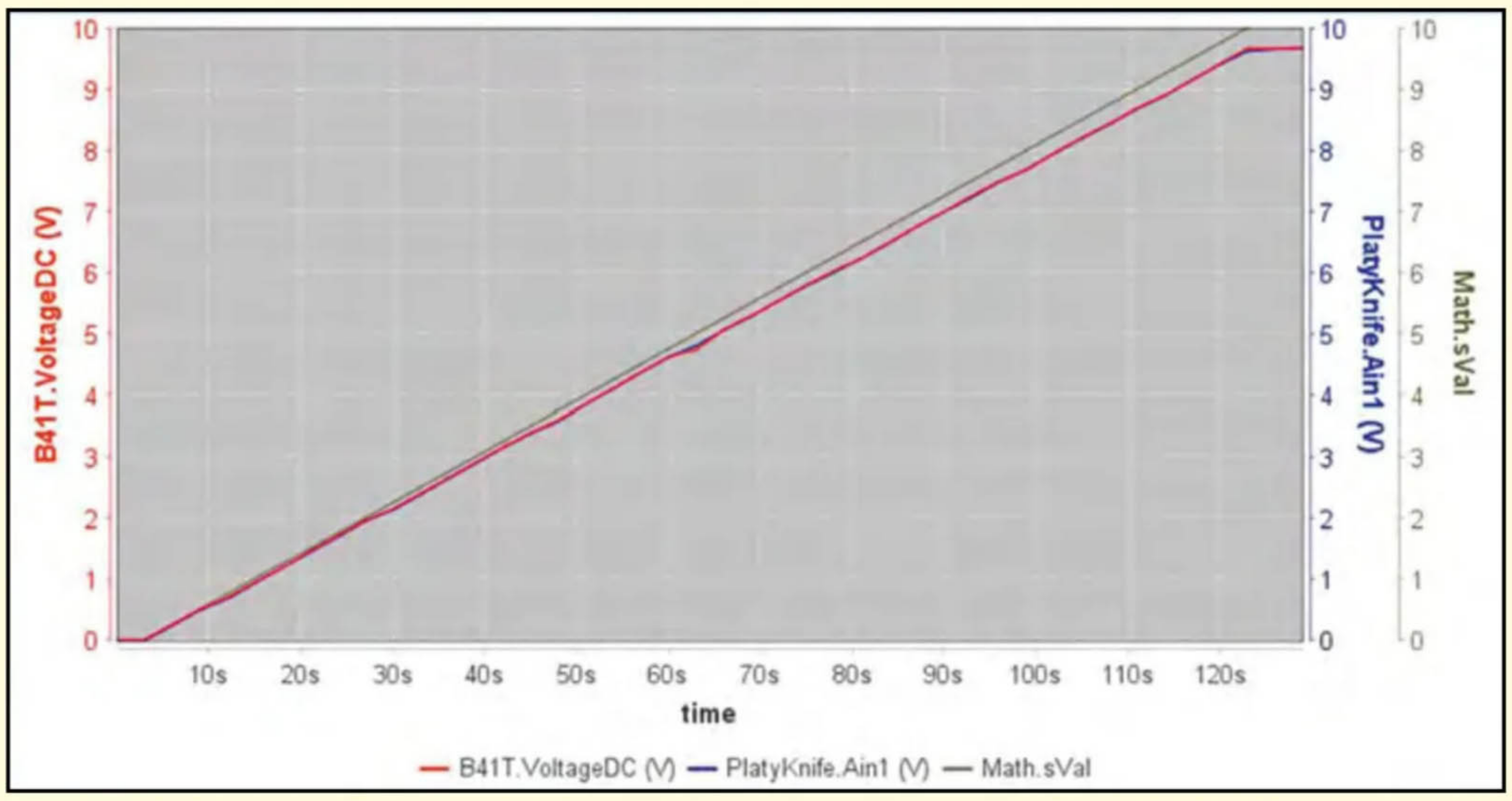
To demonstrate how much this helps, compare Screen 1 to Screen 9, which is the same measurement without the averaging. The analogue input measurement (blue trace) also has some unevenness, representing a variation of a few counts between ADC readings.

These scripts were run many times during the project's development, saving time and avoiding transcription errors. Even at a modest hourly rate, the time saved more than equalled the entire cost of the Swiss Army Knife's components.

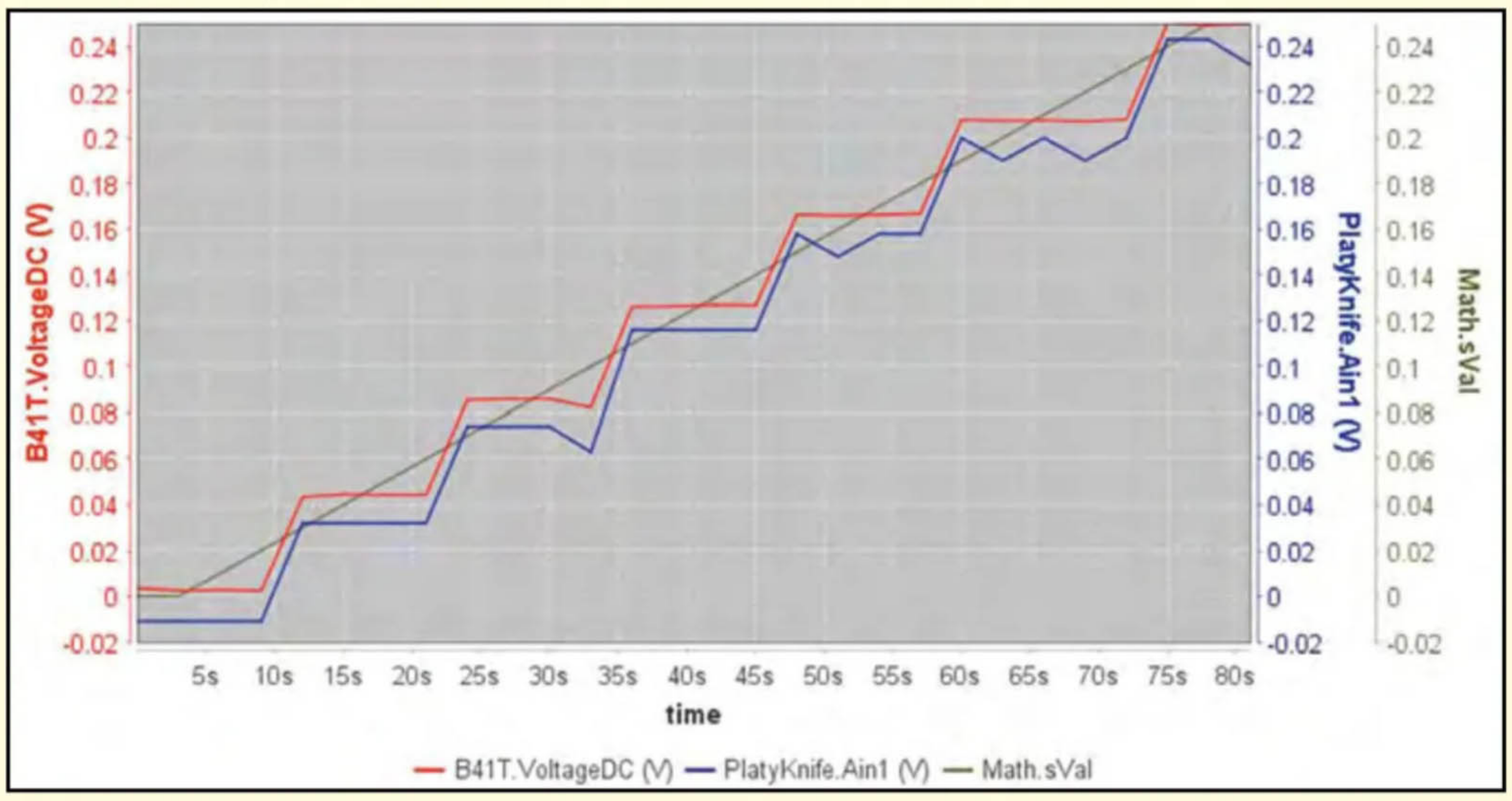


```
; ADC & DAC voltage tracking test
; create a control variable that can be logged
=globalvar sVal=0
; set initial value, let it settle and wait until value is logged
=sVal=0.0
PlatyKnife::SOUR:A1 0.00
#delay 3
; don't log commands and log values every 3 seconds
#logcmds 0
#log 3
#hasLogged
; each iteration: update analogue output and wait for logging
#while (sVal<10.2)
    PlatyKnife::SOUR:A1 (sVal)
    #hasLogged
    =sVal=(sVal+0.25)
#endwhile
#hasLogged
#log 0
```

A TestController script I used to test the Swiss Army Knife. After setting up the initial values, the analogue output value is incremented by 0.25V until the limit is reached. Each cycle waits for the log entry to be written before updating to the next value.



Screen 8: the tracking of the analogue input and output against the value measured on a B41T multimeter over the complete output range of 0-10V. Note that while the analogue input and multimeter readings track well, the analogue output had not yet been calibrated and is low (Math.sVal is the analogue output setting).

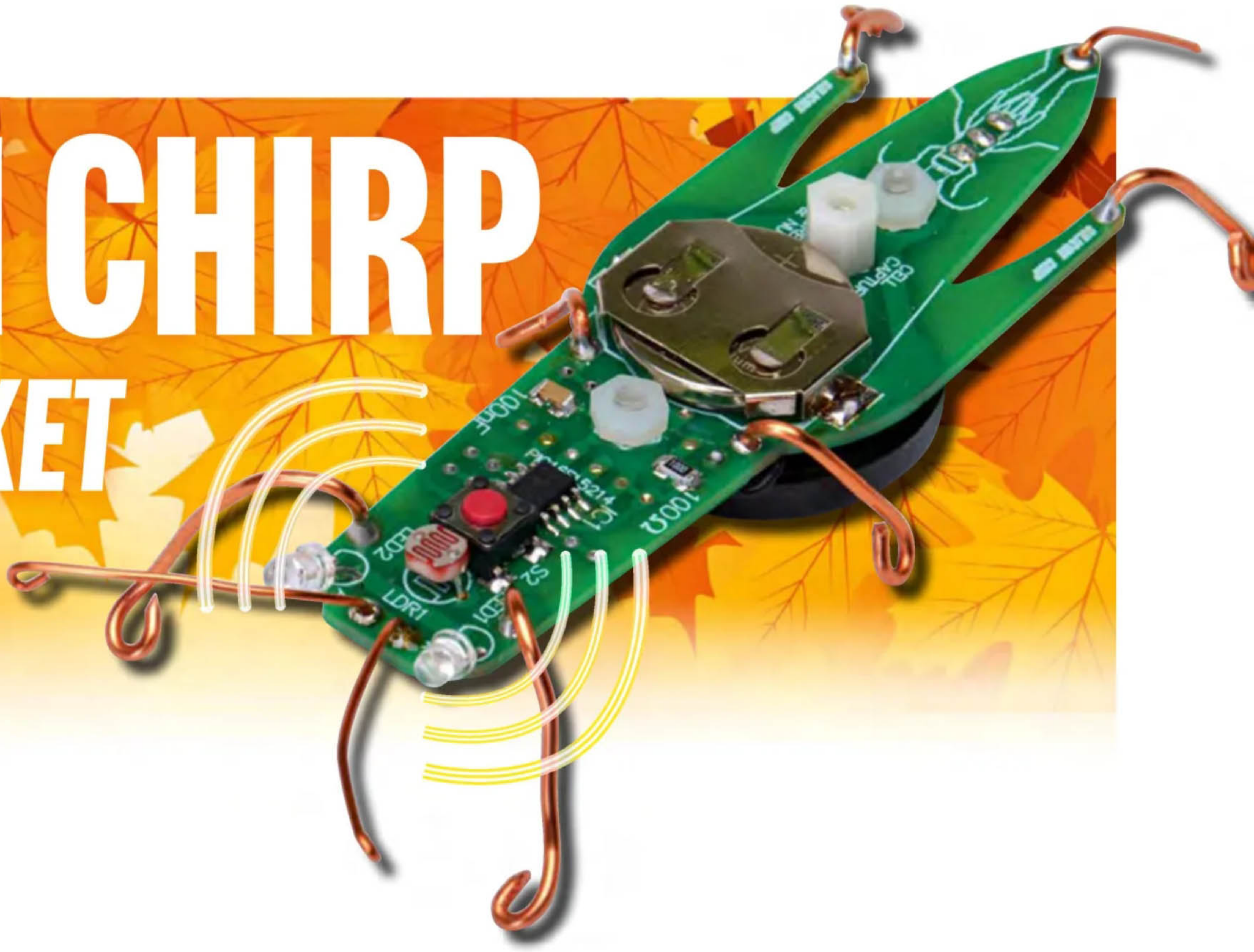


Screen 9: the performance at the low end of the analogue scale without input sample averaging. You can see the DAC steps of just over 40mV. The ADC's offset is around -1mV and tracks the multimeter well at low voltages. The 1 LSB jitter seen here is all but eliminated by the averaging done by the firmware.

SILICON CHIRP

THE PET CRICKET

BY JOHN CLARKE



This pet cricket will keep you company; it only needs to be fed occasionally and won't run away. Keep it for yourself or play a prank on a family member or friend by hiding it in their room. When they switch the lights off, they will get a bit of a surprise!

Crickets, frogs and canaries tend to be organic, made from tried-and-tested construction materials such as DNA and proteins. Until now, that is. *Silicon Chirp*, the electronic cricket, sounds like a real cricket. Not only is this project fun, it totally (and unexpectedly for a cricket) mimics frog and canary sounds. With very few parts, it is easy and fun to build.

Silicon Chirp loves to sing in the dark and happily chirps away, much to the annoyance of others. When disturbed by light, (s)he ceases, thus hiding its whereabouts until darkness falls again. But (s)he does not immediately begin to chirp again when darkness falls. That could take up to 40 seconds.

And as you enjoy the peace and when all thoughts of an annoying cricket drift away...chirping starts. And so begins the hunt for that pesky critter. Catching its glinting eyes in the dark, you are faced with a predicament: remain petrified and unable to move, or face that terrifying sight!

When the novelty of cricket sounds wears off, it can be changed to a frog, croaking in the dark. Or, for something completely different, change the sound to a singing canary to brighten your day.

Why call this critter *Silicon Chirp*? The name comes from the fact that the workings to produce the cricket sound are based upon silicon DNA. Also, it produces a chirping sound. Hence the name: *Silicon Chirp*.

As mentioned, *Silicon Chirp* can produce the sound of a frog or canaries and, of course, a cricket shape is inappropriate when making these alternative sounds. We considered having three separate PCBs with different shapes, but swapping parts from one board to the other seemed like overkill.

Then again, the Bower Bird still looks like a bird, even when making sounds like a chainsaw or a car alarm.

So, this cricket is a keen ventriloquist, mimicking the sounds of other animals while remaining in the cricket shape. It's so talented that its legs and mouth don't even move while making those sounds! You could place a frog or bird toy near *Silicon Chirp* to make the ventriloquism seem all the more real.

For the cricket, most components are mounted on *Silicon Chirp*'s back, with its eyes being 3mm red LEDs. The piezo transducer that produces the sounds is slung under the PCB abdomen. Six legs are fashioned from thick 1.25mm copper wire, while the two antennae and ovipositor (tail) are made from a thinner gauge wire.

Cricket sounds

Crickets produce their iconic chirping sounds by rubbing a coarse section of one wing against a scraper on the other. This process is called 'stridulation'; it's a bit like running a stick along a picket fence or old-fashioned washboard.

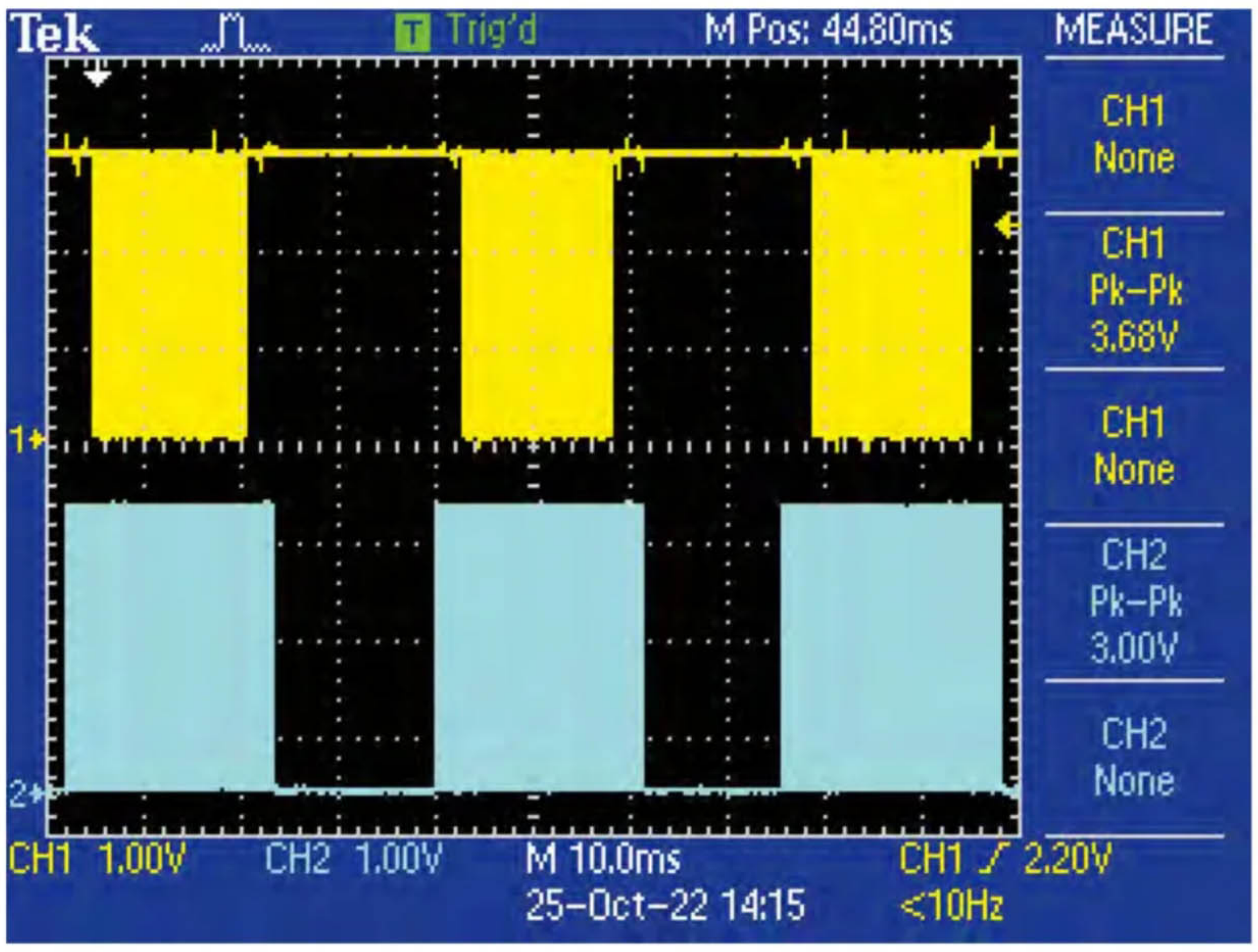
Typically, the sound a cricket produces comprises three closely spaced chirps, followed by a longer gap, then another three and so on (ie, they have a particular pattern or cadence).

A typical cricket chirp comprises four bursts of a 4kHz tone, each lasting for around 50ms. The spacing between each chirp is also about 50ms, while the separation between each triplet is around 250ms.

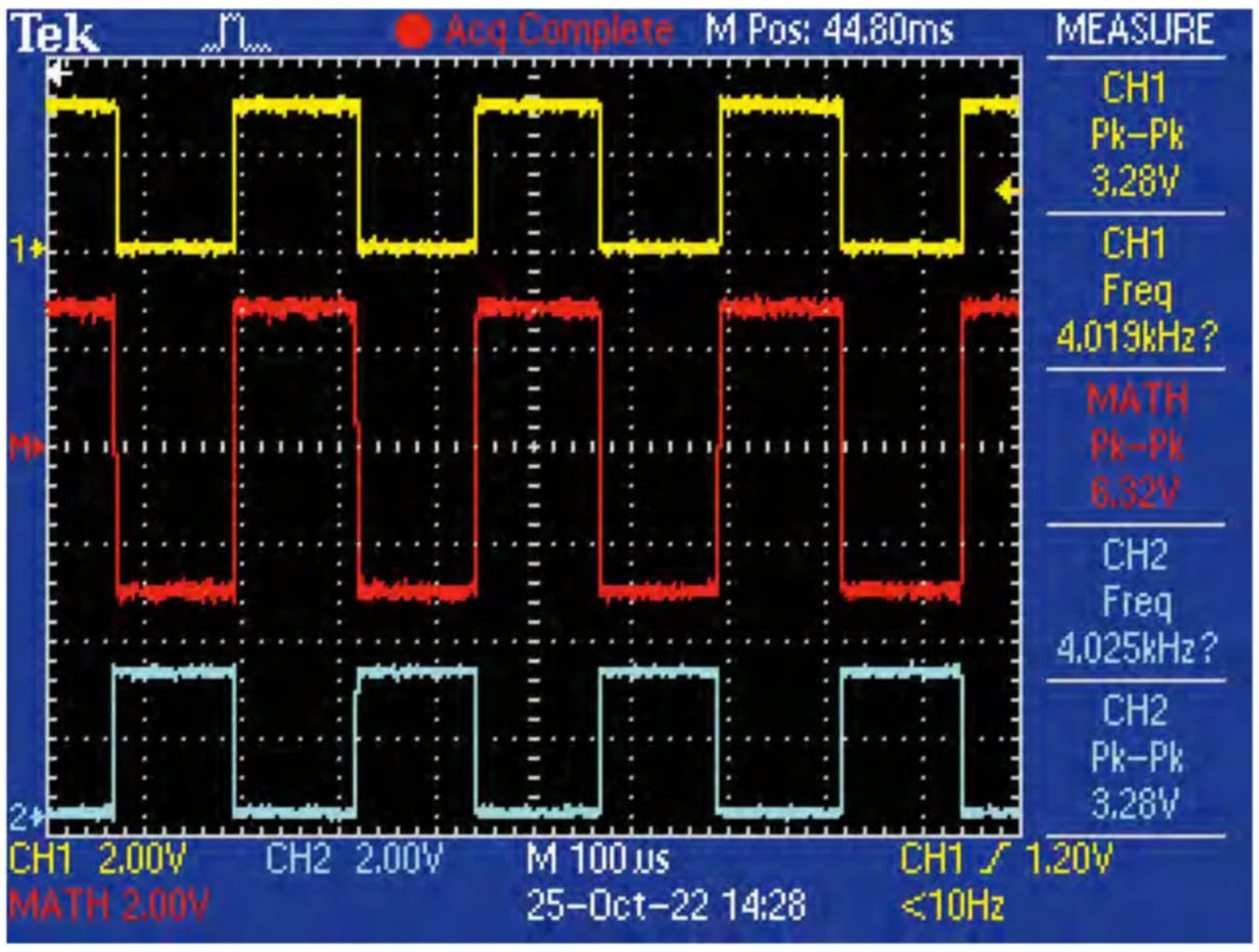
These periods are not precise and do vary a little. However, the tone of the chirp does not appear to vary by any noticeable degree.

Features and Specifications

- Looks and sounds like a cricket
- Also has the option to produce frog or canary sounds
- Flashing red eyes
- Can be set to only operate in the dark (or light, in canary mode)
- Low current draw from 3V lithium coin cell
- Current draw: 0.4µA while dormant, 0.48-1.7mA during chirps



Scope 1: cricket-like chirping is simulated by driving the piezo with groups of three signal bursts spaced apart by around 20ms. These groups have much longer silent periods in between them.



Scope 2: a close-up of the drive to the piezo, showing how the 3V peak-to-peak square wave signals from the RA0 and RA1 outputs (yellow and cyan traces) combine to produce a 6V peak-to-peak square wave across the transducer (red trace).

Silicon Chirp follows the same pattern, with triplets of 4kHz bursts, each separated by a longer gap. However, we found that driving a piezo transducer with three 20ms bursts at 4kHz and with 20ms gaps between them produced the most authentic cricket sound, even though the 20ms periods are different from that of an actual cricket.

The screen grab in Scope 1 shows the *Silicon Chirp's* cadence as measured by an oscilloscope.

To act like a real cricket, the chirp rate must vary slightly rather than being at precise intervals. So *Silicon Chirp's* chirping periods vary randomly over a limited range. In other words, they aren't always exactly 20ms long or spaced apart by precisely 20ms. The variations in the periods provide a more natural cadence and help prevent the simulated cricket chirp from sounding fake or artificial.

Frog sounds are produced similarly but with a different cadence to the cricket. For *Silicon Chirp*, frog sounds comprise a set of 10 chirps, 10ms long with 2ms gaps.

This is followed by a 30ms gap and then another set of three chirps. The ten and three groups are separated by a delay of 200-1200ms that varies irregularly. The frequency of the chirps is set at around 2kHz.

The canary sounds have been divided into three types, designated A, B and C. Song A sounds like a typical canary, while Song B simulates a Fife canary. Song C is a selection that comprises various single phrases produced by these birds.

The canary sings at random. Each song is repeated between two and 27 times with a 2.4 to 17-second gap between them. There is an extended gap between each series of repeated songs, between 80 seconds and nine minutes. Like the cricket and frog, the bird songs are produced by varying the frequency, volume and length of bursts of pulse trains applied to the piezo.

The sound volume is varied by changing the pulse width of the signals applied to the piezo transducers. Narrow pulses generate a lower volume, while the wider pulses make more sound. The maximum (loudest) pulse width equates to a duty cycle of 50%.

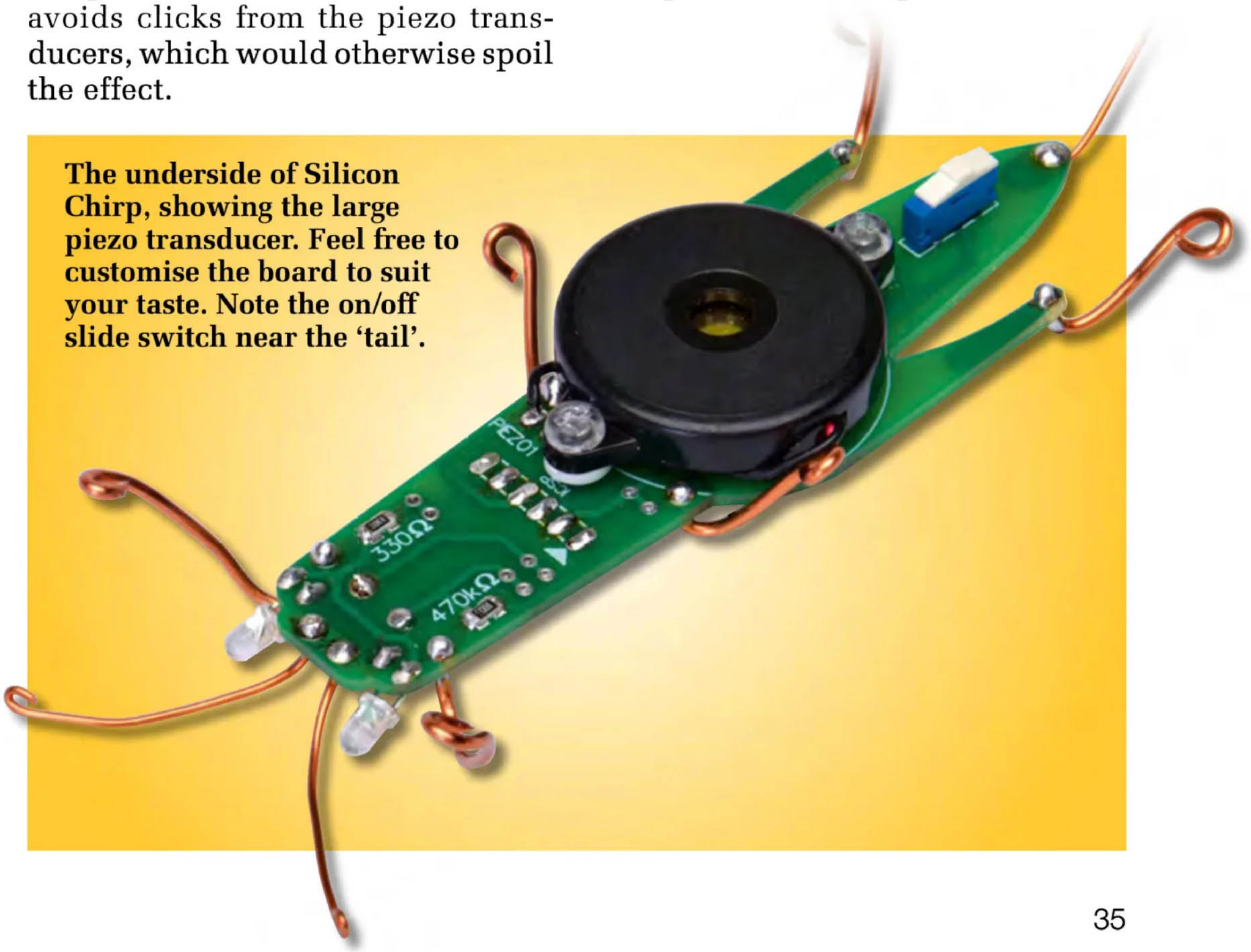
Each chirp starts at the minimum pulse width, increasing to the required volume level over time. Similarly, the pulse width is reduced to zero over a short interval when a chirp or tweet is about to end. This avoids clicks from the piezo transducers, which would otherwise spoil the effect.

Unlike crickets and frogs, which tend to make noise when it's dark, bird sounds occur mainly when it is light. So the light/dark detection is inverted for the canary.

Circuit description

The complete *Silicon Chirp* circuit is shown in Fig.1. It's based around microcontroller IC1, a PIC16F15214-I/SN, powered by a 3V lithium cell, switched via slide switch S1. IC1 does not draw much current, typically only about 400nA while it is dormant. This rises to between around 480µA to 1.7mA while making a noise.

Diode D1 is included as a safety measure to prevent damage to IC1 should the cell be inserted incorrectly. The correct polarity is with the positive side up, but the cell



The underside of Silicon Chirp, showing the large piezo transducer. Feel free to customise the board to suit your taste. Note the on/off slide switch near the 'tail'.

holder will accept the cell in either possible orientation.

With the positive side down, the cell will be shorted out by contact with the sides and top spring contacts. However, during insertion, there could be a brief period when there is no contact with the cell holder sides, so the circuit could be supplied with a reversed voltage polarity that could damage IC1.

Diode D1 clamps any reverse voltage to a low level. The cell will lose some capacity if left connected in reverse for more than a few seconds, but that's better than damaging the IC.

IC1's power supply is bypassed with a 100nF capacitor and runs using its internal 4MHz oscillator. When dormant, this oscillator is shut down (ie, in 'sleep mode') to save power. A 'watchdog' timer starts running to wake IC1 periodically (at approximately four-second intervals). During this period, the current consumption is typically less than 1µA.

During the waking period, IC1 checks the ambient light level on the light-dependent resistor, LDR1. Most of the time, the RA5 output (pin 2) of IC1 is set high (3V), so there is no current flow through the 470kΩ resistor and the LDR to minimise the current drain.

When IC1 is awake, it sets the RA5 output low (0V) and the LDR forms a voltage divider with the 470kΩ resistor across the 3V supply. The RA4 digital input (pin 3) monitors the voltage across LDR1.

In darkness, the LDR resistance is high (above 5MΩ), so the voltage at the RA4 input is more than 2.7V due to the voltage divider action of the LDR and the 470kΩ resistor. This voltage is detected as a high level by IC1. With sufficient light, the LDR resistance drops below 10kΩ, so the voltage divider produces a low level of 63mV or less at the RA4 input.

The thresholds for the RA4 input are 20% of the supply voltage for low and 80% of the supply for high. It is a Schmitt-trigger input, so once it exceeds the high threshold, the voltage must drop below 20% of the supply to switch to low. Similarly, once detecting a low, the voltage must go above 80% of the supply before a high level is indicated.

That ensures there is no rapid switching between high/low state detection when the voltage is between these thresholds.

Driving the piezo transducer

IC1's RA0 and RA1 digital output pins (pins 7 and 6) drive the piezo transducer that produces the chirps. The piezo is driven in bridge mode, connected across these two outputs, which increases the AC voltage to produce a louder sound.

When RA0 is driven high, the RA1 output is taken low; when the RA0 output is low, RA1 is high. In one condition, there is +3V across the piezo transducer and in the other, -3V, producing a 6V peak-to-peak square wave. This is shown in the Scope 2 screen grab.

Scope 2 is a close-up of the 4kHz drive waveform fed to the piezo sounder. Channels 1 and 2 (yellow and cyan traces) are the signals applied at either end of the piezo transducer, while the red trace shows the total. So, while each end of the piezo is driven by a 3.28V peak-to-peak waveform, there is double that voltage produced across the piezo.

A 100Ω resistor limits the peak current into the transducer's capacitive load immediately after the outputs switch.

LED1 and LED2 are driven via the RA2 (pin 5) and RA5 digital outputs with 330Ω current-limiting resistors. These LEDs are driven alternately on and off while the piezo transducer is driven. When RA5 is low and RA2 high, LED1 is lit, while when RA5 is high and RA2 is low, LED2 lights.

Note that RA5 is also used to drive the LDR (LDR1) to monitor the ambient light level. When driving RA5 low for light measurement, RA2 is also set low, so the LEDs are off. Similarly, when the LDR is off (RA5 high), RA2 is also brought high to keep the LEDs off.

Pushbutton switch S2 changes the sound produced from cricket to frog or canary. IC1 detects when S2 is closed by monitoring digital input RA3 (pin 4). When S2 is pressed, the voltage at that pin goes to 0V. When the switch is open, the internal pull-up at RA3 keeps that input level high. The S2 switch closure is only checked during power-up; changing the sound can only be done then.

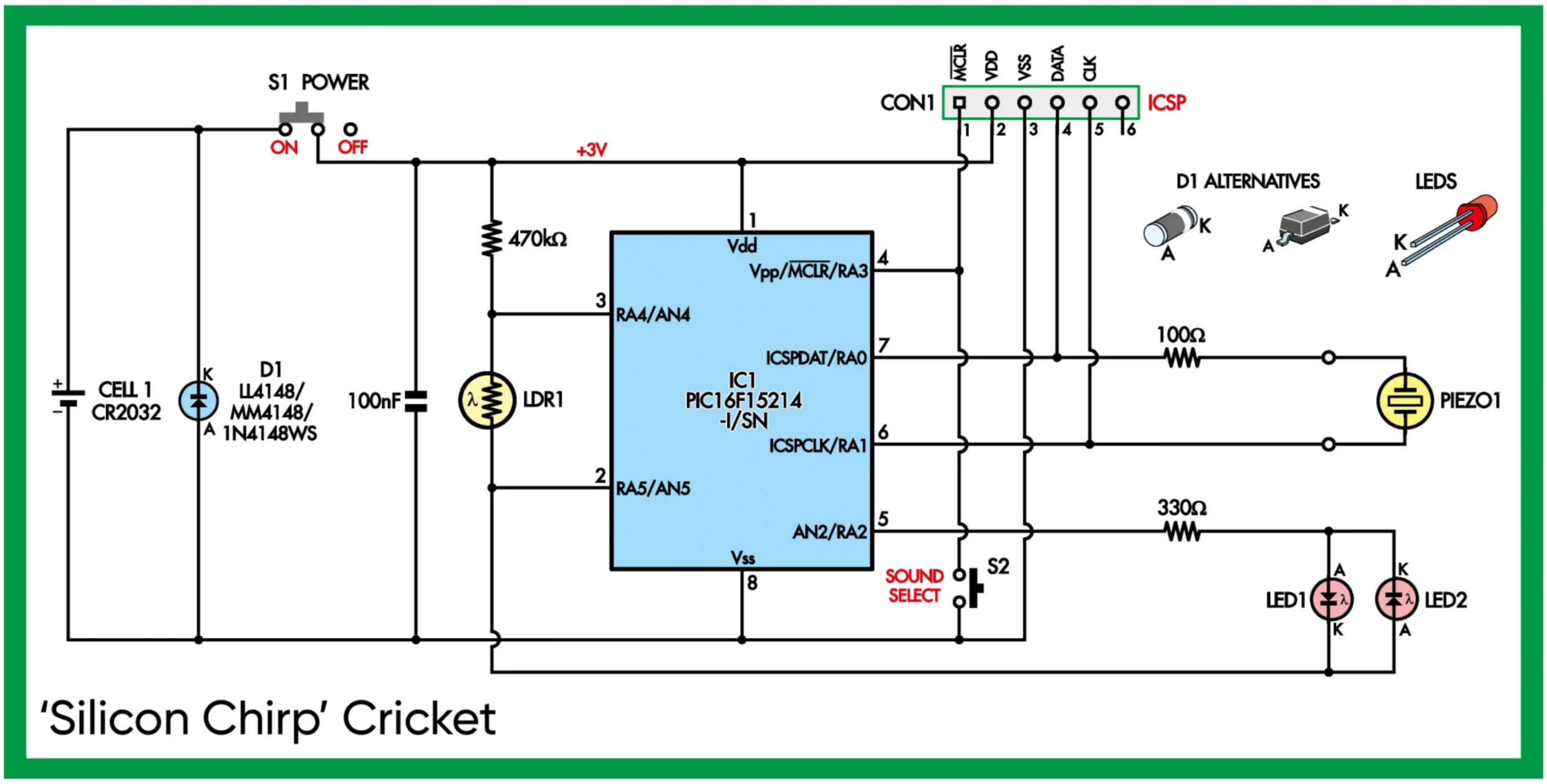


Fig.1: Silicon Chirp is controlled by 8-bit PIC16 microcontroller IC1. Slide switch S1 applies power from the coin cell. It then uses LDR1 to sense the light level and, depending on what it finds, produces sounds by driving the piezo transducer from its pin 6 and 7 digital outputs while flashing the eye LEDs via the pin 2 and pin 5 digital outputs.

Construction

Silicon Chirp is built on a double-sided, plated-through PCB coded 08101231 that measures 94 × 30.5mm available form the *PE PCB Service*. Wire legs are soldered to this PCB so it 'stands up' like a real cricket. These wires and the other parts are shown in Fig.2 and Fig.3.

CON1 is the in-circuit serial programming (ICSP) header, which is needed to program a blank micro.

Screen printing for this is on the underside of the board (for aesthetic reasons); however, it needs to be installed from the top since only the underside of the PCB has exposed pads for soldering. The top layer pads are masked, also for aesthetic reasons. Ideally, you should remove the ICSP connector after programming, as real crickets do not tend to have a programming connector.

Begin by installing the surface-mounting microcontroller, IC1. You will need a soldering iron with a fine tip, a magnifier and good lighting. The use of flux paste during soldering is advised, in which case you don't necessarily need a very fine soldering iron tip.

Solder IC1 to its PCB pads by first placing it with the pin 1 locating dot to the top left, positioning the IC leads over their corresponding PCB pads. Then tack-solder a corner pin and check that the IC is still aligned correctly. If you find that it needs to be realigned, remelt the soldered connection and gently nudge the IC into correct alignment.

Once alignment is correct, solder all the IC pins and refresh that initial joint. Any solder that runs between the IC pins can be removed with solder paste and the application of solder-wicking braid.

Continue construction by installing the resistors. They are printed with a code indicating their values, which is likely to be '1000' or '101' for 100Ω, '3300' or '331' for 330Ω and '4703' or '474' for 470kΩ. These are in 'scientific notation' where the last digit indicates the number of zeros to add to the first few digits to give a value in ohms.

Diode D1 can be installed next. Remember that it is polarised, so take care to orient it correctly, with the cathode stripe facing away from the centre of the PCB. There is sufficient pad area to allow Mini-MELF (SOD-80) or SOD-323 package diodes to be soldered in. Alternatively, an axial-leaded 1N4148 could be used with the leads at each end bent back by 180° to allow soldering to the PCB pads.

The 100nF capacitor is fitted next, and since it is an unpolarised part it can be positioned either way round.

We installed slide power switch S1 on the PCB's underside. You could place this on top if you prefer. The on position for the switch is when the slider is toward the front of the cricket. You can also mount pushbutton switch S2 now by soldering its four pins.

Parts List – Silicon Chirp Cricket

1 double-sided, plated-through PCB coded 08101231, 94 × 30.5mm
  available fro the PE PCB Service
1 CR2032 surface-mounting coin cell holder (CELL1) **[BAT-HLD-001]**
1 CR2032 3V lithium cell
1 SPDT micro slide switch (S1) **[Jaycar SS0834]**
1 SPST surface-mounting tactile pushbutton switch (S2)
  [Altronics S1112A, Jaycar SP0610]
1 30mm diameter 4kHz wired piezo transducer (PIEZO1)
  [Altronics S6140, Jaycar AB3442]
1 45k-140kΩ light dependent resistor (LDR1)
  [Altronics Z1619, Jaycar RD3480]
3 M3 × 10mm panhead machine screws (metal or plastic)
1 M3 × 6.3mm tapped Nylon spacer (or two M3 hex nuts)
2 Nylon or polycarbonate M3 hex nuts
2 TO-220 insulating bushes (eg, from TO-220 insulating kits)
  [Altronics H7110, Jaycar HP1142]
1 6-way header with 2.54mm pitch (CON1; **optional**, for programming IC1)
1 200mm length of 1.25mm diameter enamelled copper wire (for legs)
1 100mm length of 1mm diameter enamelled copper wire
  (for antennae and ovipositor)

Semiconductors
1 PIC16F15214-I/SN 8-bit microcontroller programmed
  with 01810123A.hex, SOIC-8 (IC1)
2 3mm red LEDs (LED1, LED2)
1 LL4148, MM4148 or 1N4148WS (or 1N4148; see text) SMD diode,
  Mini-MELF (SOD-80) or SOD-323 **[Altronics Y0161/Y0164A]**

Capacitors
1 100nF 50V X7R SMD M3216/1206 size

Resistors *(all M3216/1206 size 1%)*
1 470kΩ　　1 330Ω
1 100Ω

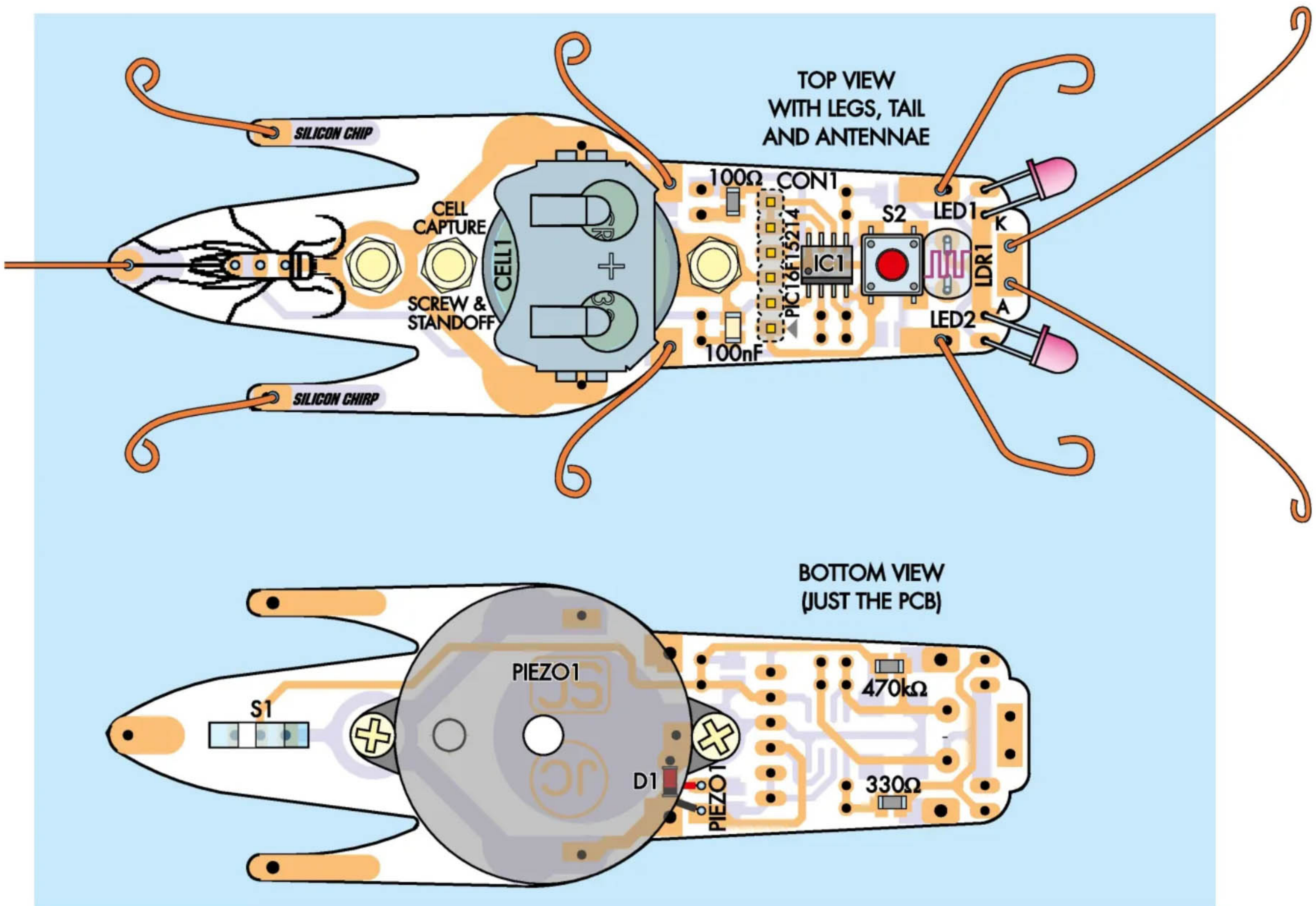
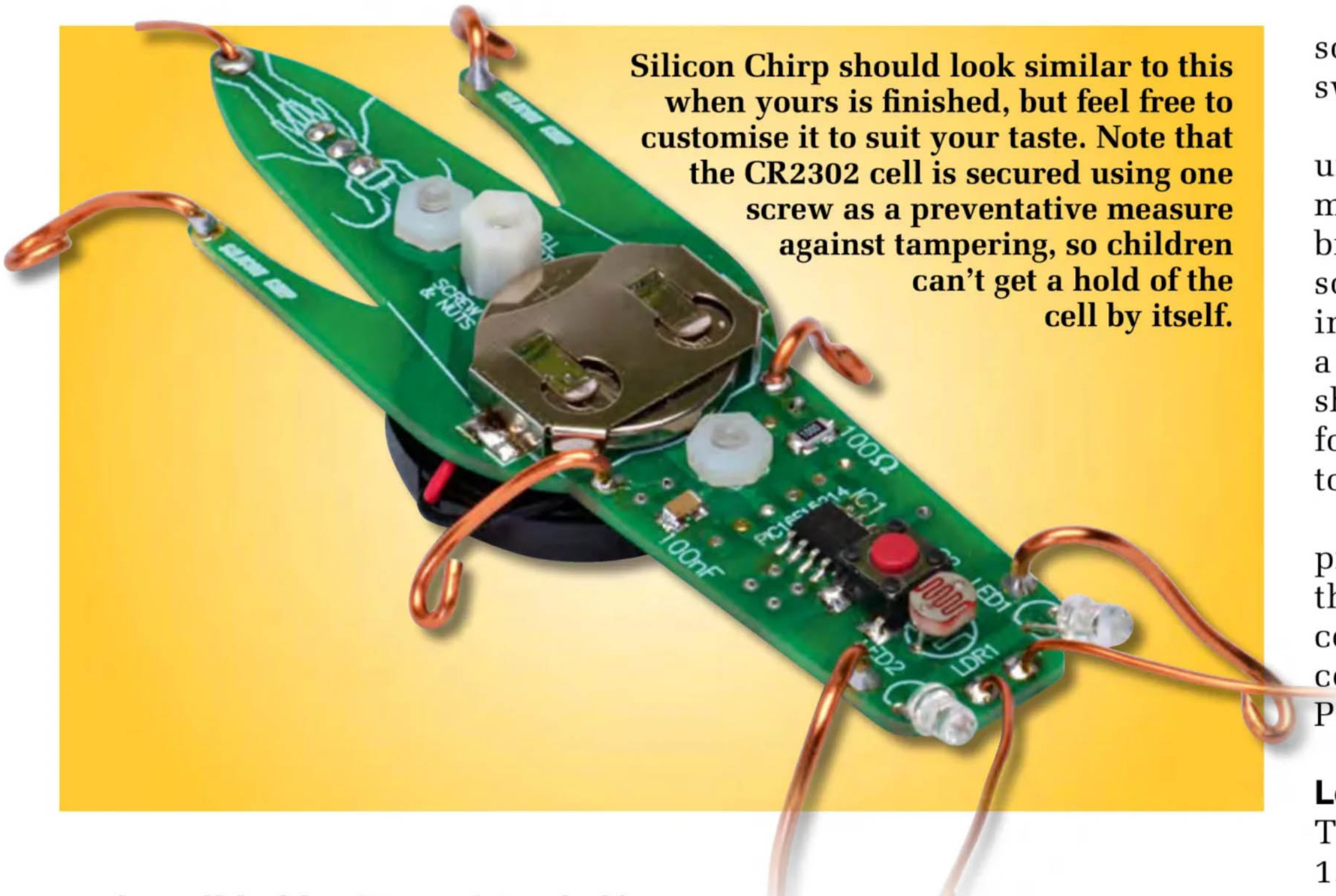


Fig.2 and Fig.3: Silicon Chirp is pretty easy to build. Simply place the components as shown here but note that the piezo transducer is wired and mounted over reverse polarity protection diode D1. That diode, IC1 and the LEDs are polarised and must be soldered the right way around; the other components are not polarised.


Silicon Chirp should look similar to this when yours is finished, but feel free to customise it to suit your taste. Note that the CR2302 cell is secured using one screw as a preventative measure against tampering, so children can't get a hold of the cell by itself.

The cell holder (CELL1) is a half-shell type and its body makes contact with the positive side of the cell. A tinned copper area on the PCB completes the cell holder and provides for the negative connection to the cell. It must be fitted with the cell entry toward the rear of the cricket so that the cell capture screw prevents small children from removing it.

This is to comply with my local safety standards (Australian Standard AS/NZS ISO 8124.1:2002), where toys for children three years and younger must have any batteries (and/or cells) secured in a compartment by a screw. Alternatively, where there is no compartment screw used, there must be two simultaneous independent movements to open the battery compartment.

While *Silicon Chirp* is not really a project for small children, it could be used in a household with children who could potentially swallow button or coin cells, which poses a serious hazard (see the warning panel for details).

For our project, cell removal is blocked by a 10mm M3 machine screw inserted from the PCB's underside and secured on top with an M3-tapped Nylon spacer. When tightened, the spacer cannot be removed by hand and stops the cell from being removed. An alternative to the standoff is to use two M3 nuts, with the top one used as a lock nut, tightened against the other.

Mount LED1 and LED2 so that the top of the dome of each LED is raised off the PCB by about 10mm. This provides enough lead length so they can be bent to about 30° above the PCB plane and outward about 10° from the centre line, as shown in Fig.2 and the photos. Make sure the longer lead of each LED (the anode) is inserted in the 'A' position on the PCB.

Mount the LDR about 5mm above the PCB surface, with its face sitting horizontally. This component is not polarised and can be installed either way around.

The piezo transducer is mounted on the underside of the PCB, supported on TO-220 insulating bushes that are used as spacers to raise the transducer from the PCB. This leaves room for the cell capture screw and diode to fit between the PCB and piezo. The piezo transducer is secured with two 10mm M3 machine screws and two Nylon or polycarbonate nuts.

You will need to drill out the mounting holes on the piezo unit to a 3mm diameter to suit the M3 screws. The nuts will not fit in the room provided on the piezo transducer mounting lugs, so the screws need to enter from the piezo transducer side. The insulating bushes can then be slipped onto the screw shafts, followed by the piezo transducer, then the Nylon or polycarbonate nuts.

We use plastic nuts because a metal nut will short out the cell if used at the end of the cell nearest to IC1. That's because the PCB hole and surrounding track are connected to ground, while the metal of the cell holder connects to the cell positive. To avoid any potential confusion and prevent the wrong type of nut from being placed at each point, we have specified both piezo-securing nuts as plastic.

Note that to remove the cell capture screw when the cell needs to be replaced, one of these piezo mounting screws will need to be removed so that the piezo transducer can be swung out of the way.

Solder the piezo wires to the underside of the PCB at the positions marked 'PIEZO1'. You could instead bring them to the top of the PCB and solder them through the corresponding top holes, although that will look a bit messy. The wires will need to be shortened, but leave sufficient length for the piezo to swing out of the way to access the cell capture screw.

The piezo transducer wires will probably be red and black, although the transducer is not a polarised component. It does not matter which colour wire goes to the two piezo PCB pads.

Legs and antennae

The legs can be fashioned from 1.25mm-diameter enamelled copper wire. Each front leg is 40mm long, while the mid and rear legs are each 30mm. These can be as simple or as fancy as you like. The cricket shape printed at the rear of the PCB shows the general leg shape we used, as do Fig.2 and the photos.

Bend the legs so that *Silicon Chirp's* PCB is above the platform it sits on. Form the feet into small loops so that the sharp ends of the wires are not exposed.

Where the legs are soldered to the PCB, you will need to scrape off the enamel insulation (eg, using a sharp hobby knife or fine sandpaper) before you can solder them.

Make up the two antennae using 40mm lengths of 1mm-diameter enamelled copper wire and the ovipositor (tail) with a 20mm length of the same. Once in place, curl the two antenna wires into shape by running a thumbnail along the inside of the radius, with your index finger on the outside.

Now install the CR2032 cell in its holder and switch on power with S1. If all is well, the LEDs will momentarily flash after about three seconds to acknowledge that power has been connected to the circuit.

An acknowledgement by a brief flashing of the LEDs also occurs when a low light level is detected for the cricket and frog, or when a high light level is detected for the canary. Low light can be simulated by covering over the LDR, or a higher light level by shining light onto the LDR.

Silicon Chirp will begin chirping after a delay of about 10 seconds, providing the low light level remains for the whole time.

To program the PIC, you can download the firmware (01810123A.hex) from the April 2024 page of the *PE* website: **https://bit.ly/pe-downloads** Additionally, as mentioned previously, ICSP (in-circuit serial programming) header CON1 will need to be installed. One of the piezo transducer leads may need to be disconnected, or one end of the 100Ω resistor, to allow programming.

Changing the sound

Changing from cricket to frog to canary and back is performed by holding switch S2 while switching power on via S1. Continue to hold S2 until you see the eyes flashing. They will flash once for the cricket, twice for the frog and three times for the canary. To change to the next selection, continue holding S2 for two seconds until the eyes flash to show the next selection.

When you see the selection you want, release S2. The selected sound is stored in flash memory, so that selection remains even if powered off and on again. It only changes when S2 is pressed during power-up.

Note that the frog sounds are best expressed with the piezo transducer close to a flat surface to emphasise lower frequencies. The canary sounds run through a repertoire before switching off when darkness is detected, so they won't necessarily stop as soon as the light goes away.

Warning: small cell

This design uses a small lithium cell that can cause severe problems if swallowed, including burns and possible perforation of the oesophagus, stomach or intestines. Young children are most at risk. Read the information sheet at www.schn.health.nsw.gov.au/fact-sheets/button-batteries on the dangers of button cells.

Ensure that the cell is kept secure using the cell capture screw and Nylon spacer as specified, tightened sufficiently so they cannot be undone by hand. Keep unused cells in a safe place away from children, such as a locked medicine cupboard. New cells should be kept within the original secure packaging until use.

Unfortunately, some older button-cell-powered devices not intended for children under three provide easy access to the cells. Keep these away from children or devise a method to make cell access more difficult (eg, by gluing the compartment shut).

Modifications

Silicon Chirp has a loud chirp, which can be pretty annoying! (But maybe you want that...) To reduce the volume, increase the value of the 100Ω resistor in series with the piezo transducer. Increasing it to, say, 10kΩ will reduce the apparent volume by about 50%. Higher values will provide an even lower volume, to the point where it won't chirp at all.

The light sensitivity can also be altered by changing the 470kΩ resistor value between the positive supply and the PIC's RA4 input. Increasing the resistance value (say to 1MΩ) will make the light threshold level darker. By contrast, reducing the resistance value will mean more light is required to detect daytime.

Reproduced by arrangement with
SILICON CHIP magazine 2024.
www.siliconchip.com.au

GET THE LATEST COPY OF OUR TEACH-IN SERIES AVAILABLE NOW!

Order direct from Electron Publishing

PRICE £8.99
(includes P&P to UK if ordered direct from us)

Teach-In 9 – Get Testing!

This series of articles provides a broad-based introduction to choosing and using a wide range of test gear, how to get the best out of each item and the pitfalls to avoid. It provides hints and tips on using, and – just as importantly – interpreting the results that you get. The series deals with familiar test gear as well as equipment designed for more specialised applications.

The articles have been designed to have the broadest possible appeal and are applicable to all branches of electronics. The series crosses the boundaries of analogue and digital electronics with applications that span the full range of electronics – from a single-stage transistor amplifier to the most sophisticated microcontroller system. There really is something for everyone!

Each part includes a simple but useful practical test gear project that will build into a handy gadget that will either extend the features, ranges and usability of an existing item of test equipment or that will serve as a stand-alone instrument. We've kept the cost of these projects as low as possible, and most of them can be built for less than £10 (including components, enclosure and circuit board).

PLUS! You will receive the software for the *PIC n' Mix* series of articles and the full *Teach-In 2* book – *Using PIC Microcontrollers – A practical introduction* – in PDF format. Also included are Microchip's *MPLAB ICD 4 In-Circuit Debugger User's Guide*; *MPLAB PICkit 4 In-Circuit Debugger Quick Start Guide*; and *MPLAB PICkit4 Debugger User's Guide*.

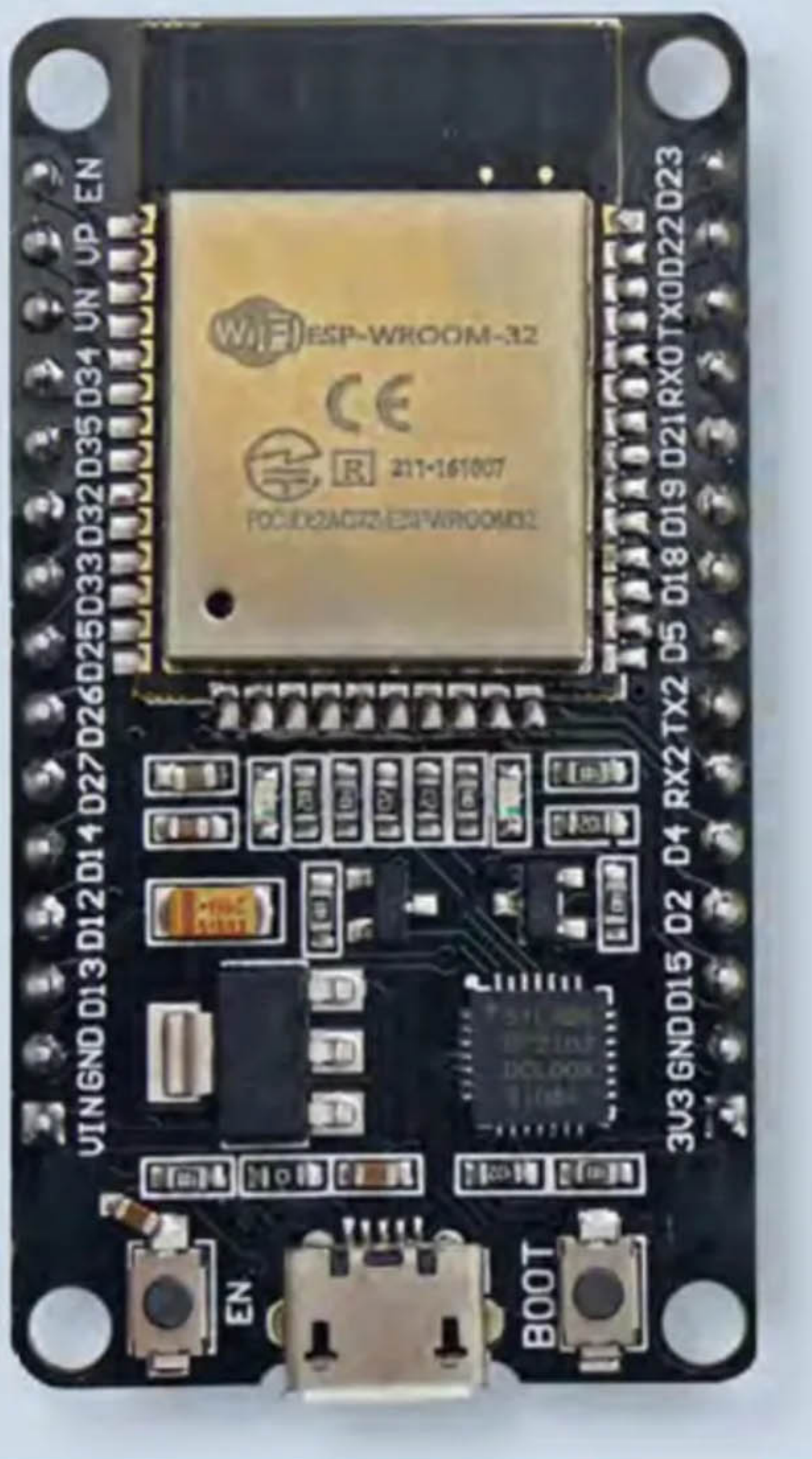
ORDER YOUR COPY TODAY: www.electronpublishing.com

Teach-In 2024

Learn electronics with the ESP32

by Mike Tooley

Part 2 – Digital input and output


ESP-WROOM-32
EN
BOOT
VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23

In last month's Part 1 of our Teach-In series we provided an overview of the ESP32 and introduced the development environment. We showed you how to use your PC to monitor the ESP32's built-in capacitive touch sensors and our practical project involved using the ESP32 in a portable emergency beacon. This month, we will be taking a more detailed look at digital I/O and showing you how to interface buttons and switches as well as how to drive loads such as LEDs, relays and sounders. We will also introduce the Serial Monitor. As part of the Arduino IDE, this handy tool provides a great way of testing your code. The learning objectives for this part are:

- Know how to use the Serial Monitor
- Know how to configure and use simple digital I/O
- Know how to increase the output capability of the ESP32.

About Teach-In

Our latest *Teach-In* series is about using the popular ESP32 module as a basis for learning electronics and coding. We will be making no assumptions about your coding ability or your previous experience of electronics. If you know one but not the other, you have come to the right place. On the other hand, if you happen to be a complete newbie there's no need to worry because the series will take a progressive hands-on approach. There will be plenty of time to build up your knowledge and plenty of opportunity to test things out along the way.

We've not included too much basic theory because this can be easily found elsewhere, including several of our previous *Teach-In* series, see:

https://bit.ly/pe-ti
https://bit.ly/pe-ti-bundle

There will be projects and challenges to help you check and develop your understanding of the topics covered each month.

Using the Serial Monitor

When testing even the most basic code you will often find that you need some means of knowing what's going on. Without having the benefit of a full user interface (such as a local keypad and display) this can be tricky, but the Arduino IDE can come to your rescue with its in-built Serial Monitor. However, before you can begin to use this useful tool you will need to initialise the Serial Monitor within your `setup()` code like this:

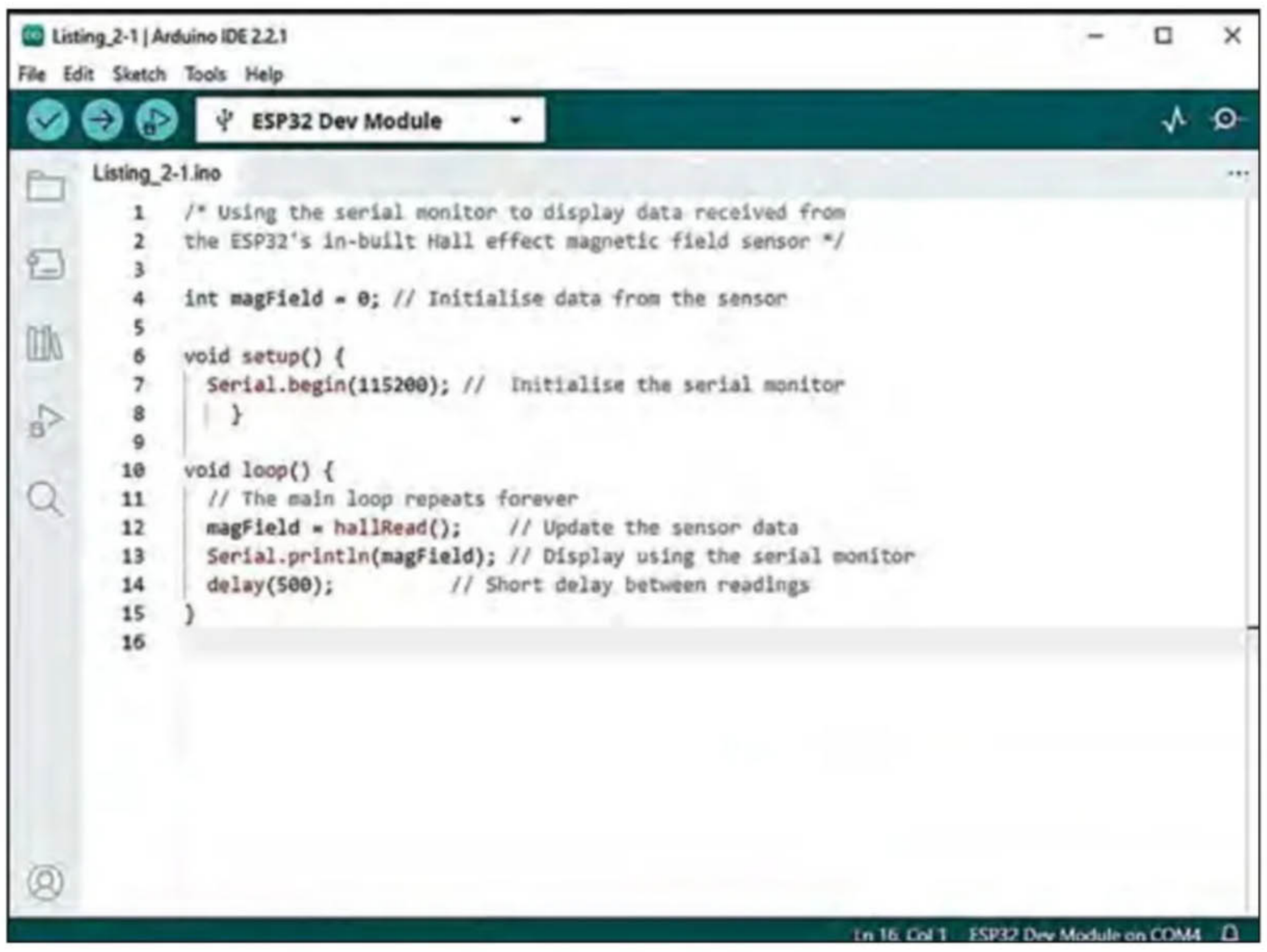
```
void setup() {
Serial.begin(9600); // Initialize the Serial
Monitor
}
```

You probably noticed the parameter that we've used in the `Serial.begin()` function. This sets the speed or data rate in bits per second (baud) that will be used for serial communication. It is typically set to either 9600 or 115200, but other baud rates can be selected if required. An optional second argument configures the data, parity and stop bits. The default is eight data bits, no parity and one stop bit. This should work fine with the Arduino IDE and many other applications, but you may need to change it for different external hardware. For example, the following code sets the serial UART to 9600 bits per second with eight data bits, even parity and two stop bits:

```
void setup() {
Serial.begin(9600, SERIAL_8E2); // Initialize
                                   the Serial Monitor
}
```

To send a message to the Serial Monitor you can use the `Serial.print` and `Serial.println()` functions. The only difference between these functions is that the latter adds a newline character at the end of the printed string. Here's an example:

```
Serial.print("Pin selected = ");
```



```
/* Using the serial monitor to display data received from
the ESP32's in-built Hall effect magnetic field sensor */

int magField = 0; // Initialise data from the sensor

void setup() {
  Serial.begin(115200); //  Initialise the serial monitor
  }

void loop() {
  // The main loop repeats forever
  magField = hallRead();     // Update the sensor data
  Serial.println(magField); // Display using the serial monitor
  delay(500);           // Short delay between readings
}

```

Fig.2.1. Code for testing the ESP32's built-in Hall effect sensor.

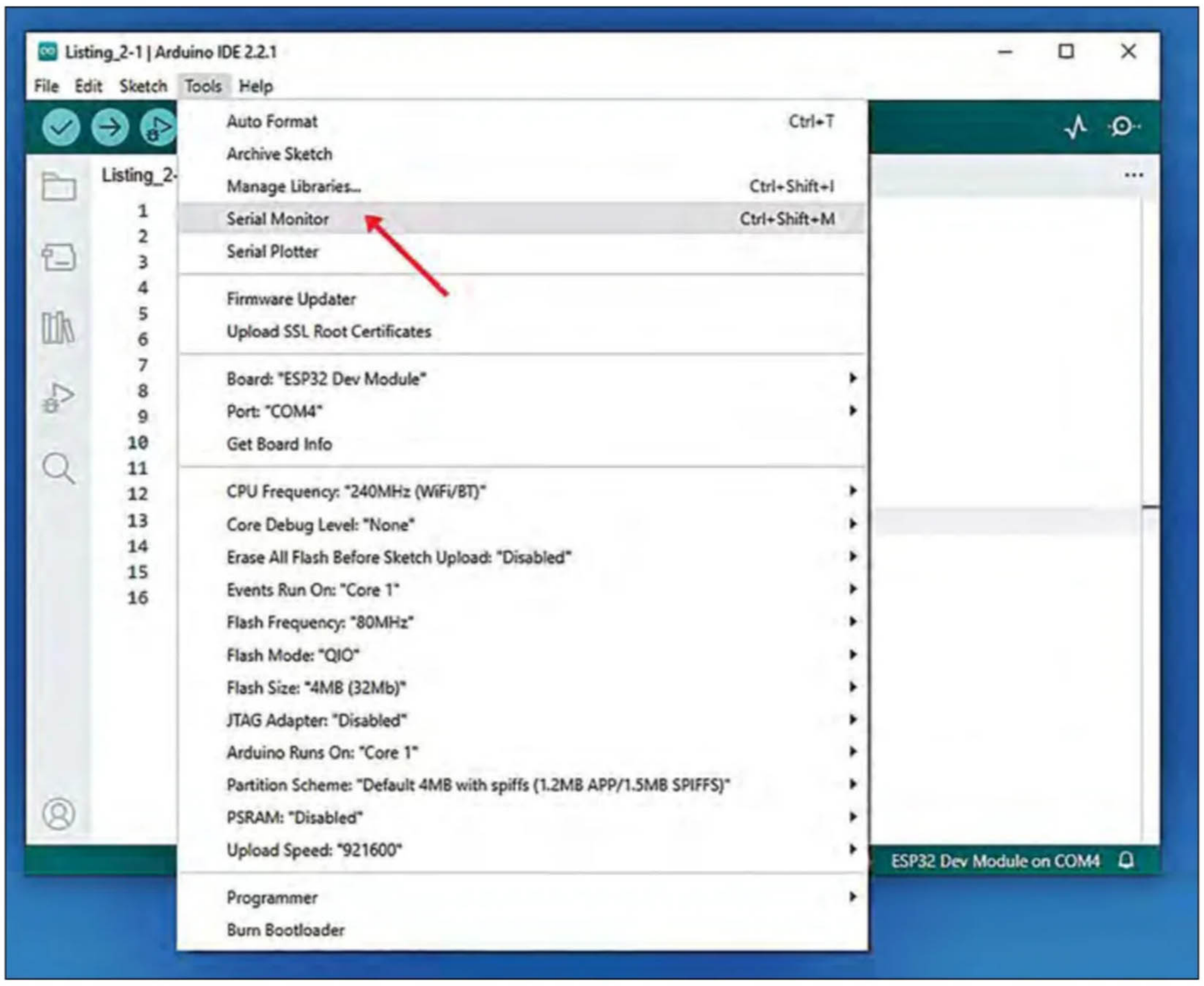

Listing_2-1 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Listing_2-
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board: "ESP32 Dev Module"
Port: "COM4"
Get Board Info
CPU Frequency: "240MHz (WiFi/BT)"
Core Debug Level: "None"
Erase All Flash Before Sketch Upload: "Disabled"
Events Run On: "Core 1"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
JTAG Adapter: "Disabled"
Arduino Runs On: "Core 1"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
PSRAM: "Disabled"
Upload Speed: "921600"
Programmer
Burn Bootloader
ESP32 Dev Module on COM4

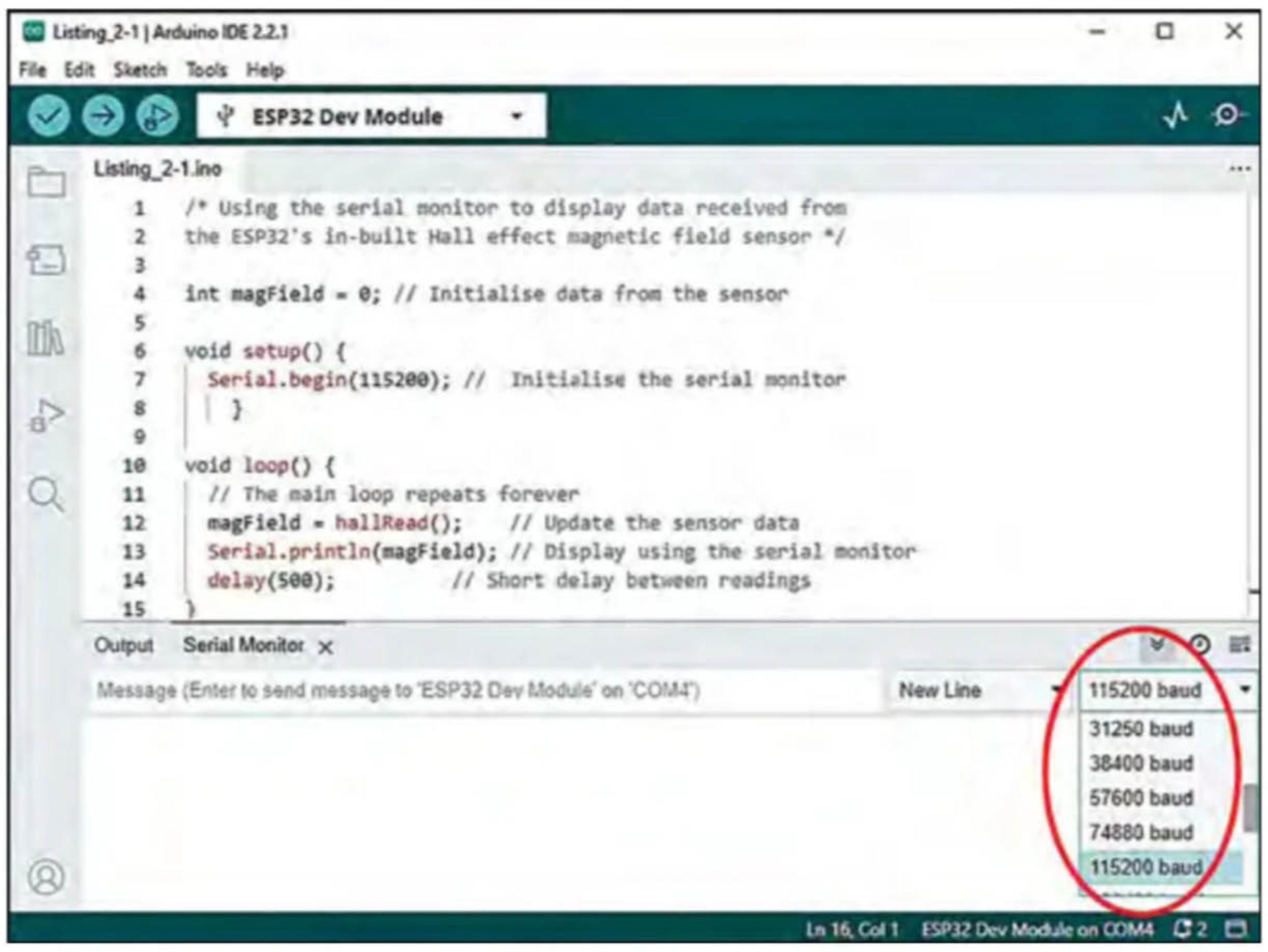
Fig.2.2. Enabling the Serial Monitor from the IDE's Tools menu.

This will print the character string enclosed in the quotes and the next item to be printed will appear *immediately after* the trailing space. So, if we wanted to add a pin number stored in the variable `selectedPin` and then *move to the next line* we would use:

```
Serial.println(selectedPin);
```

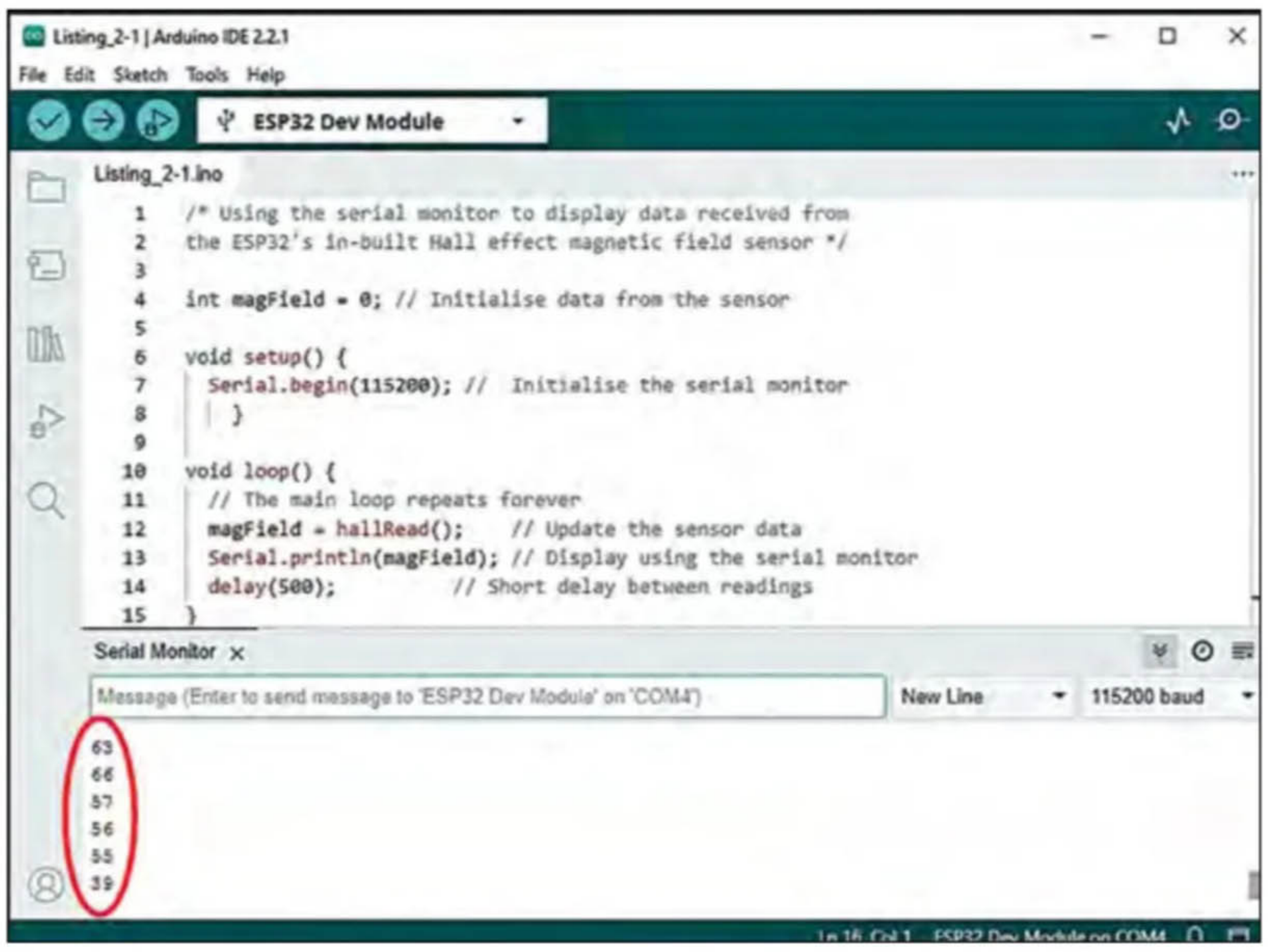
Listings 2.1 and 2.2 show how the serial monitor can be used both to receive data from, and send data to, the ESP32. The first example (Listing 2.1) makes use of the ESP32's built-in Hall effect magnetic sensor. Note that having entered or downloaded the code (Fig.2.1) you will need to enable the serial monitor from the Tools menu, as shown in Fig.2.2 and select the baud rate, as shown in Fig.2.3. When the code is executed, you will be rewarded with a list of returned data values, updated every 500ms (see Fig.2.4). The data will vary, although residual values of magnetic field should all be fairly low, but the Hall effect sensor can be tested by placing a small magnet close to the upper surface of the ESP32 module, as shown in Fig.2.5. As the permanent magnet is moved towards the ESP32 there will be a marked increase in detected field strength accompanied by a change in polarity if the direction of the field is reversed, as also shown in Fig.2.5.

The second example (Listing 2.2) shows how individual characters can be sent from the PC via the serial link. The data


Listing_2-1 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
ESP32 Dev Module
Listing_2-1.ino
Output
Serial Monitor
Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')
New Line
115200 baud
31250 baud
38400 baud
57600 baud
74880 baud
115200 baud
Ln 16, Col 1
ESP32 Dev Module on COM4

```
/* Using the serial monitor to display data received from
the ESP32's in-built Hall effect magnetic field sensor */

int magField = 0; // Initialise data from the sensor

void setup() {
  Serial.begin(115200); //  Initialise the serial monitor
  }

void loop() {
  // The main loop repeats forever
  magField = hallRead();    // Update the sensor data
  Serial.println(magField); // Display using the serial monitor
  delay(500);           // Short delay between readings
}
```

Fig.2.3. Setting the Serial Monitor's baud rate.


Listing_2-1 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
ESP32 Dev Module
Listing_2-1.ino
Serial Monitor
Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')
New Line
115200 baud
Ln 16, Col 1
ESP32 Dev Module on COM4

```
/* Using the serial monitor to display data received from
the ESP32's in-built Hall effect magnetic field sensor */

int magField = 0; // Initialise data from the sensor

void setup() {
  Serial.begin(115200); //  Initialise the serial monitor
  }

void loop() {
  // The main loop repeats forever
  magField = hallRead();    // Update the sensor data
  Serial.println(magField); // Display using the serial monitor
  delay(500);           // Short delay between readings
}
```

```
63
66
57
56
55
39
```

Fig.2.4. Returned data from ESP32's built-in Hall effect magnetic sensor.

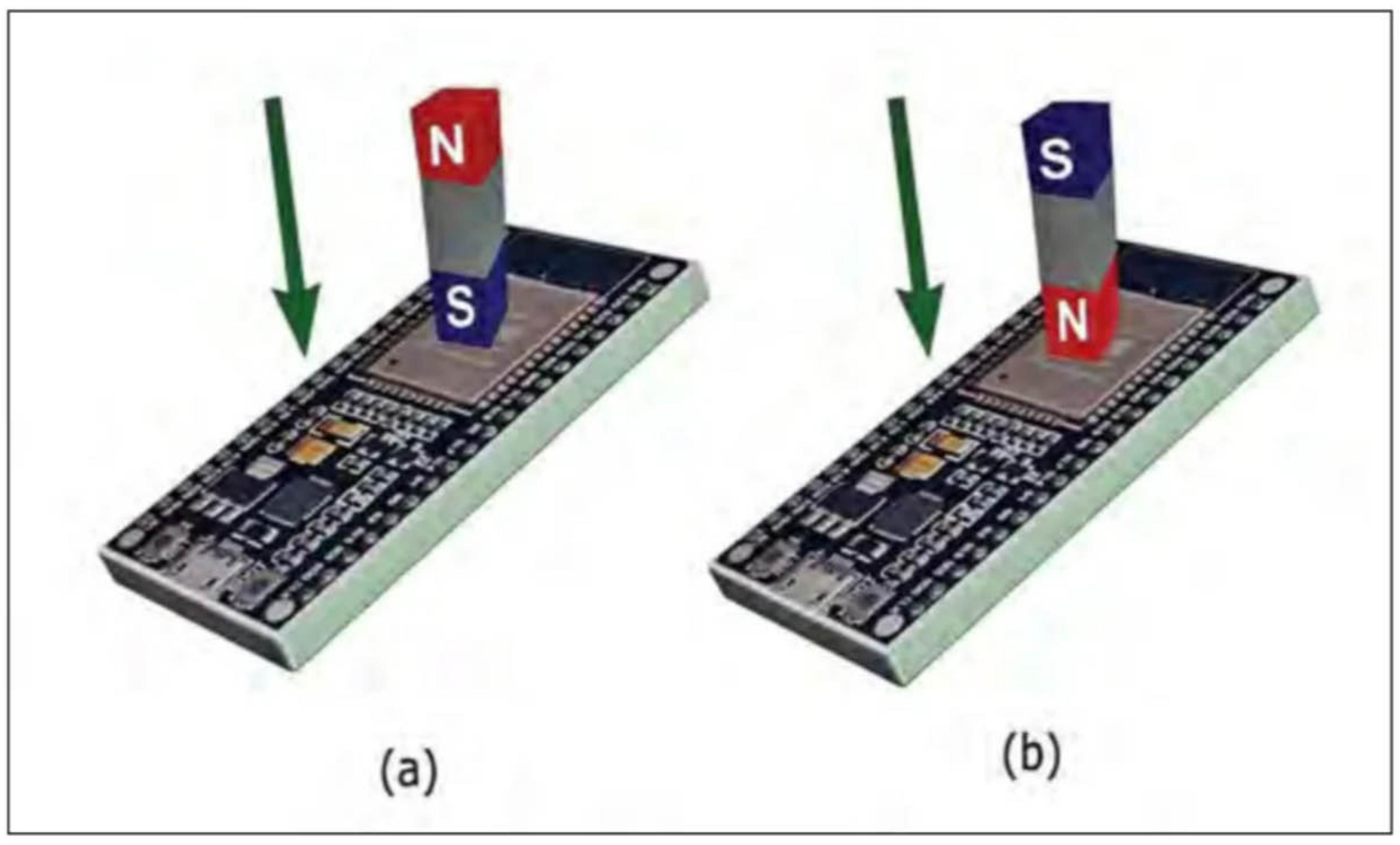


Fig.2.5. Using a small magnet to test the Hall effect magnetic sensor. Reversing the field, as shown in (a) and (b) will result in a reversal of the polarity of the returned data.

Listing 2.1 Using the serial monitor to display data from the ESP32's in-built magnetic Hall effect sensor

```
/* Using the serial monitor to display data received from
ESP32's in-built Hall effect magnetic field sensor */

int magField = 0;         // Initialise data from sensor

void setup() {
   Serial.begin(115200);  // Initialise serial monitor
}

void loop() {
   // The main loop repeats forever
   magField = hallRead(); // Update sensor data
   Serial.println(magField);     // Display using
                                 serial monitor
delay(500);               // Short delay between readings
}
```

Gotcha!

During compilation you may receive an error message informing you that ‘hallRead was not declared in this scope’. If that’s the case, it is worth checking the currently installed version of the Espressif System esp32 board library. The failure to compile seems to arise in the current ‘3.0.0 alpha’ version. To put things right you can use the IDE’s Library Manager (see *Part 1* last month) to remove the current version before installing a previous version (we used version 2.0.11 without this problem occurring).

Listing 2.2 Using the serial monitor to control the ESP32’s in-built LED

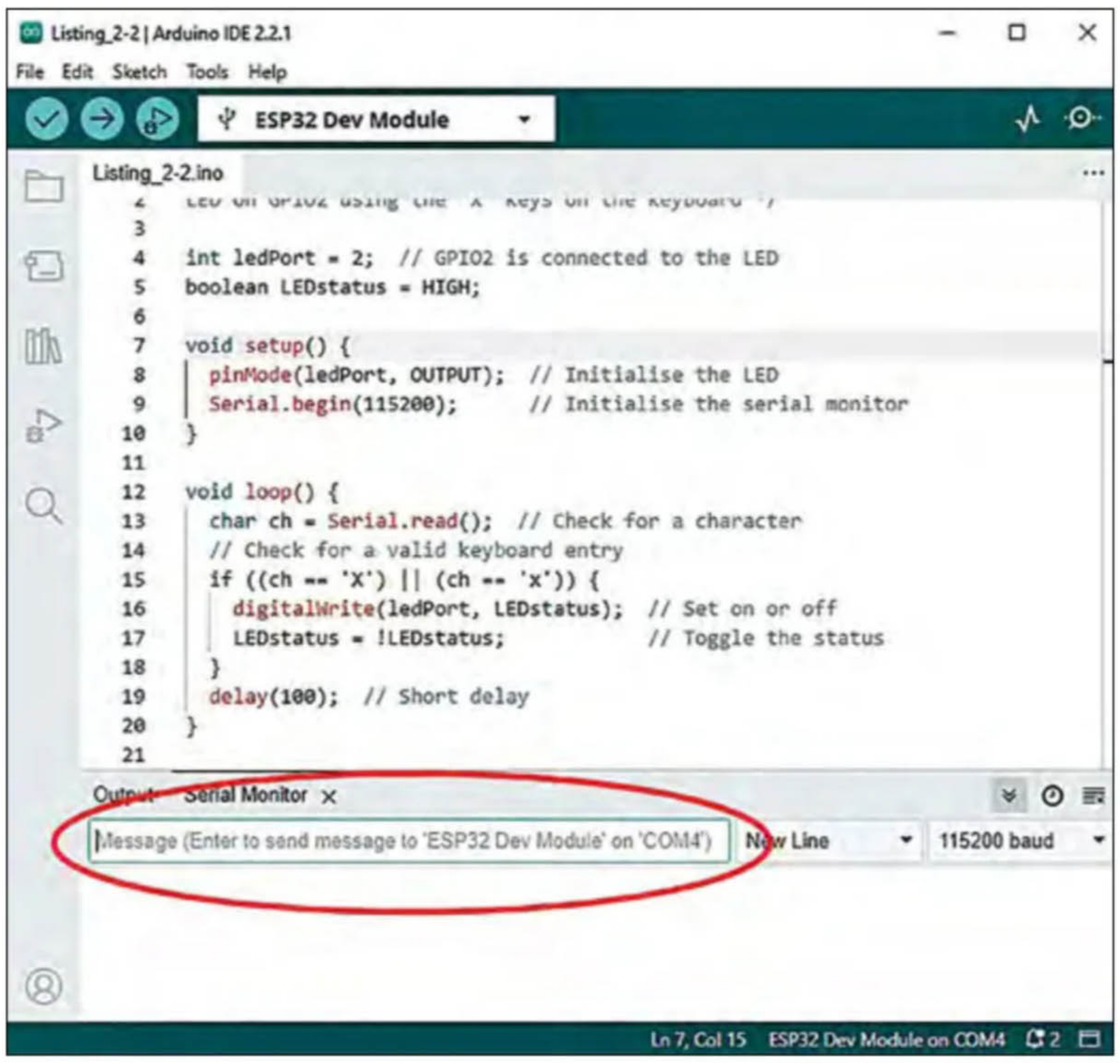
```
/* Using the serial monitor to toggle the ESP32’s in-built LED
on GPIO2 using the ‘x’ keys on the keyboard */

int ledPort = 2;                  // GPIO2 is connected to the LED
boolean LEDstatus = HIGH;

void setup() {
   pinMode(ledPort, OUTPUT);      // Initialise the LED
   Serial.begin(115200);          // Initialise the serial monitor
}

void loop() {
   char ch = Serial.read();       // Check for a character
   // Check for a valid keyboard entry
   if ((ch == ‘X’) || (ch == ‘x’)) {
     digitalWrite(ledPort, LEDstatus);     // Set on or off
     LEDstatus = !LEDstatus;      // Toggle the status
   }
   delay(100);                    // Short delay
}
```

to be sent (a single character followed by New Line) is entered into the Serial Monitor’s message field, as shown in Fig.2.6. The code will allow you to toggle the ESP32’s in-built LED (on GPIO2) ‘on’ and ‘off’ using a lower or upper case ‘X’ entered from the keyboard.

Configuring GPIO digital outputs

The output state from a GPIO line can be HIGH (approx. 3.3V) or LOW (close to 0V). The GPIO pins can source current when in the HIGH state or sink current when in the LOW state. In both cases (and to keep well within the maximum ratings for the ESP32) we recommend the current that’s sourced or sunk should *always* be less than 10mA (equivalent to a load of no less than about 330Ω). An output device (or ‘load’) can be connected so that it can be activated by either a HIGH state output or by a LOW output condition (see Fig.2.7).

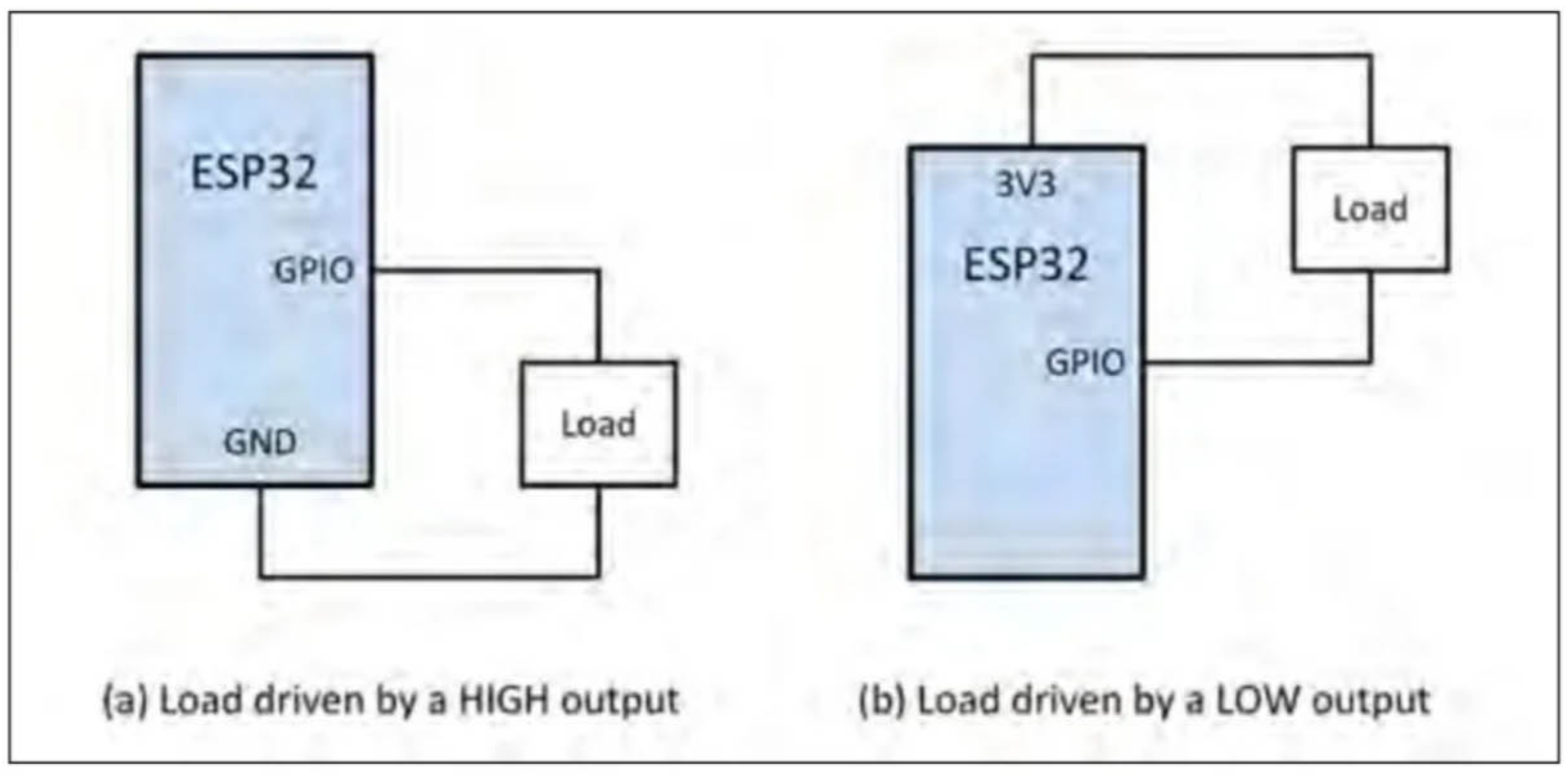
A current-limiting series resistor is required when driving an LED. The value of LED series resistor should normally be within the range 220Ω to 470Ω. For most of our examples we have used 330Ω, which results in a typical LED current of around 4mA. If more brightness is required, the value of series resistor can be reduced to 220Ω. This brings us once again to the important question of just how much current we can safely extract (source or sink) from an


Listing_2-2 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
ESP32 Dev Module
Listing_2-2.ino
Output
Serial Monitor
New Line
115200 baud
Ln 7, Col 15 ESP32 Dev Module on COM4 2

```
LED on GPIO2 using the 'x' keys on the keyboard */

int ledPort = 2;  // GPIO2 is connected to the LED
boolean LEDstatus = HIGH;

void setup() {
  pinMode(ledPort, OUTPUT);  // Initialise the LED
  Serial.begin(115200);      // Initialise the serial monitor
}

void loop() {
  char ch = Serial.read();  // Check for a character
  // Check for a valid keyboard entry
  if ((ch == 'X') || (ch == 'x')) {
    digitalWrite(ledPort, LEDstatus);  // Set on or off
    LEDstatus = !LEDstatus;            // Toggle the status
  }
  delay(100);  // Short delay
}
```

Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')

Fig.2.6. Entering character data into the Serial Monitor’s message field.



(a) Load driven by a HIGH output

(b) Load driven by a LOW output

Fig.2.7. Output loads activated by (a) HIGH and (b) LOW output states.

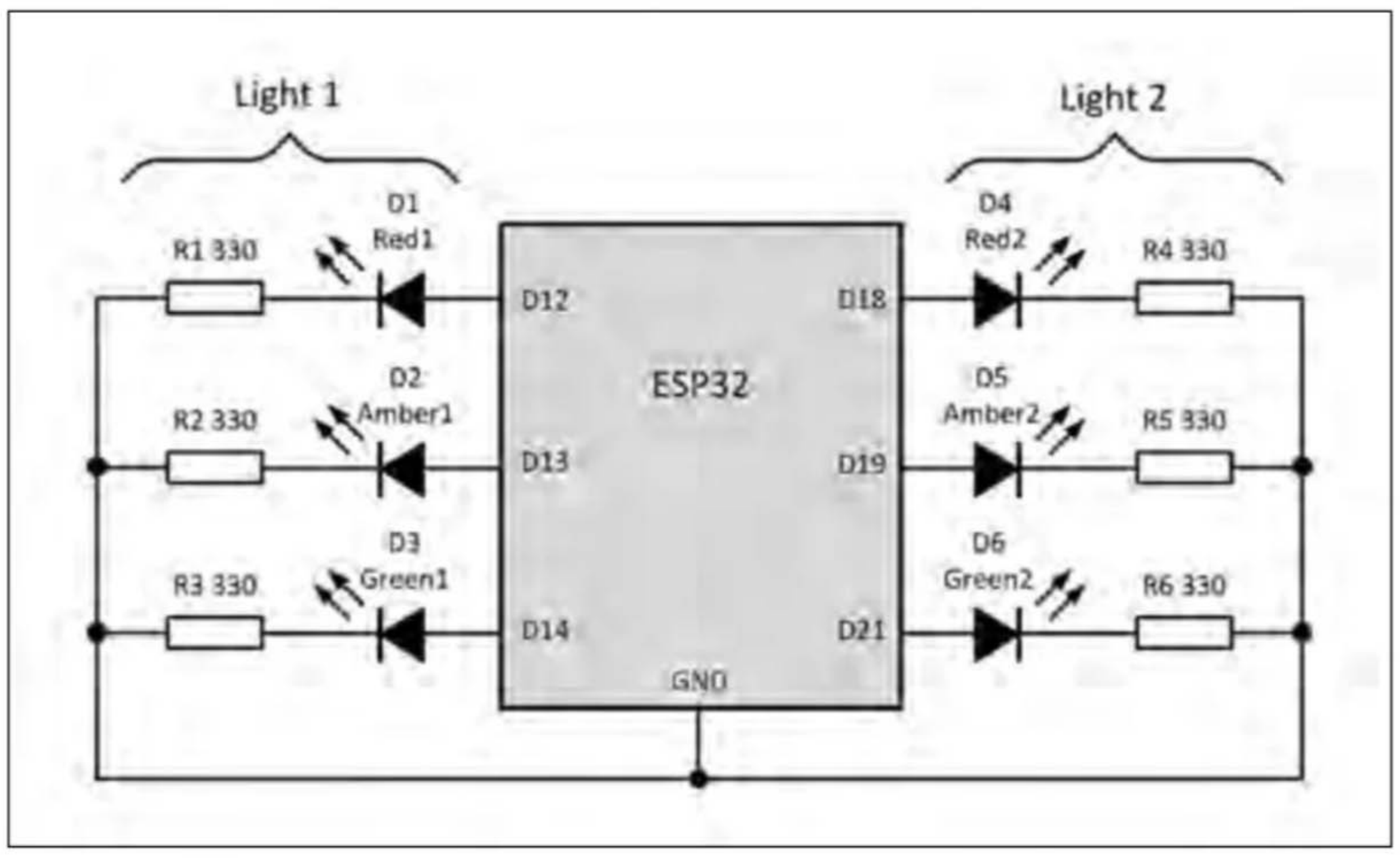


Fig.2.8. Circuit of the ESP32 traffic lights controller.

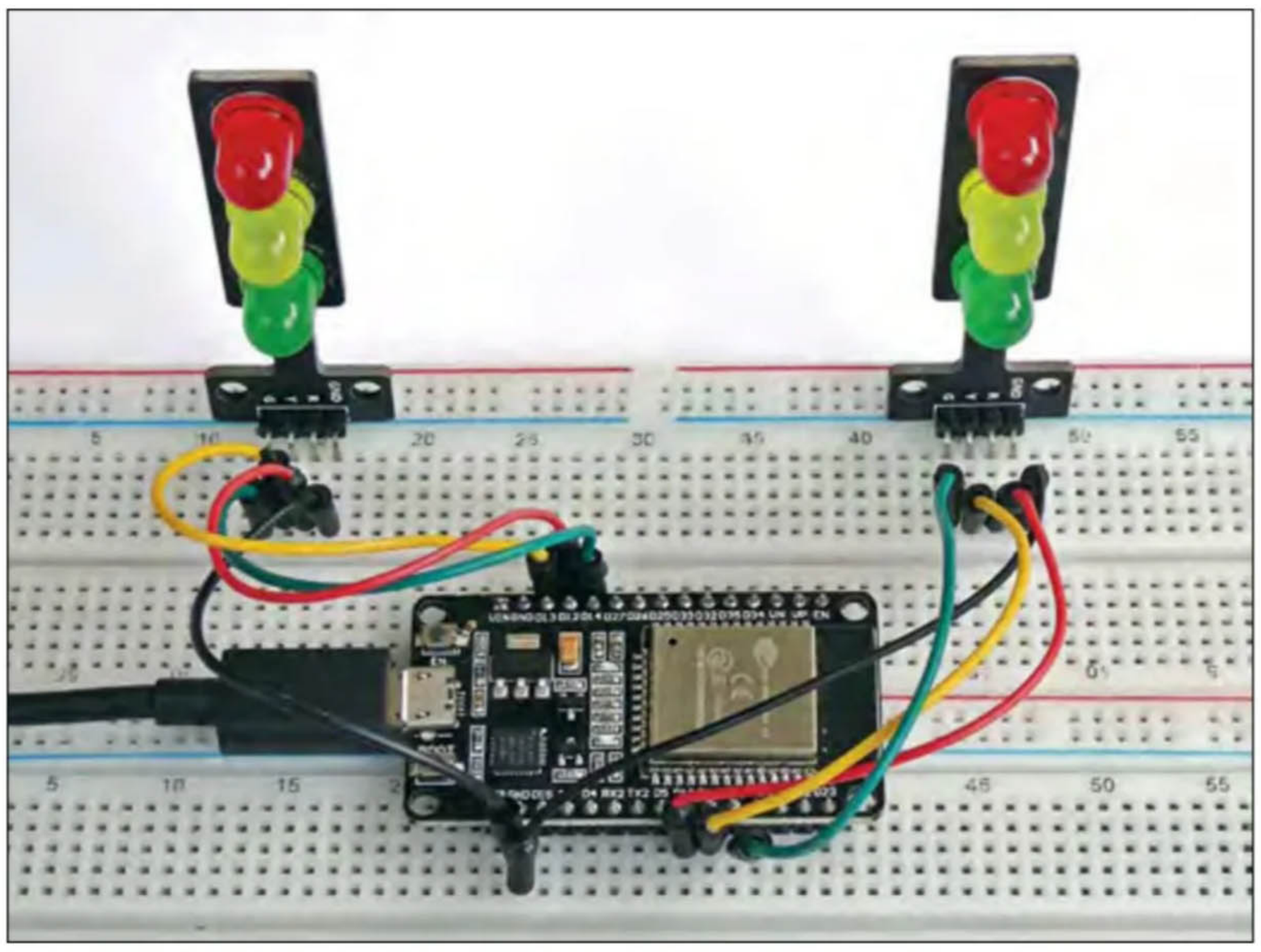


Fig.2.9. Wiring layout for the ESP32 traffic lights controller.

ESP32's GPIO pins. Although the Espressif datasheet states a maximum output drive current of 12mA, we recommend limiting the load current on any individual GPIO line to no more than 10mA, while at the same time ensuring that the total load on the ESP32's digital outputs is less than 100mA. This should give plenty of scope for most applications, but where additional current drive is required an external transistor can be used, as described later in our *Practical Project*.

Check it out!

Our *Check it out!* feature involves building a simple two-way traffic lights controller shown in Fig.2.8. Each of the two traffic lights (Light 1 and Light 2) will comprise a set of three LEDs: red, amber and green. Rather than use individual LEDs (together with series resistors) we've taken the easy option and used a ready-built mini traffic lights module fitted with red, amber and green LEDs, together with series 330Ω resistors. The module is available from several online suppliers, and it uses a common ground (GND) connection so that the individual LEDs will become illuminated when their associated input pins are taken HIGH. (Just search for '5V Mini Traffic Light LED Display'; eBay and others offer plenty of choice.)

The wiring layout for the ESP32 traffic lights controller is shown in Fig.2.9. The code shown in Listing 2.3 should be entered or downloaded from the April 2024 page of the PE website: **https://bit.ly/pe-downloads**

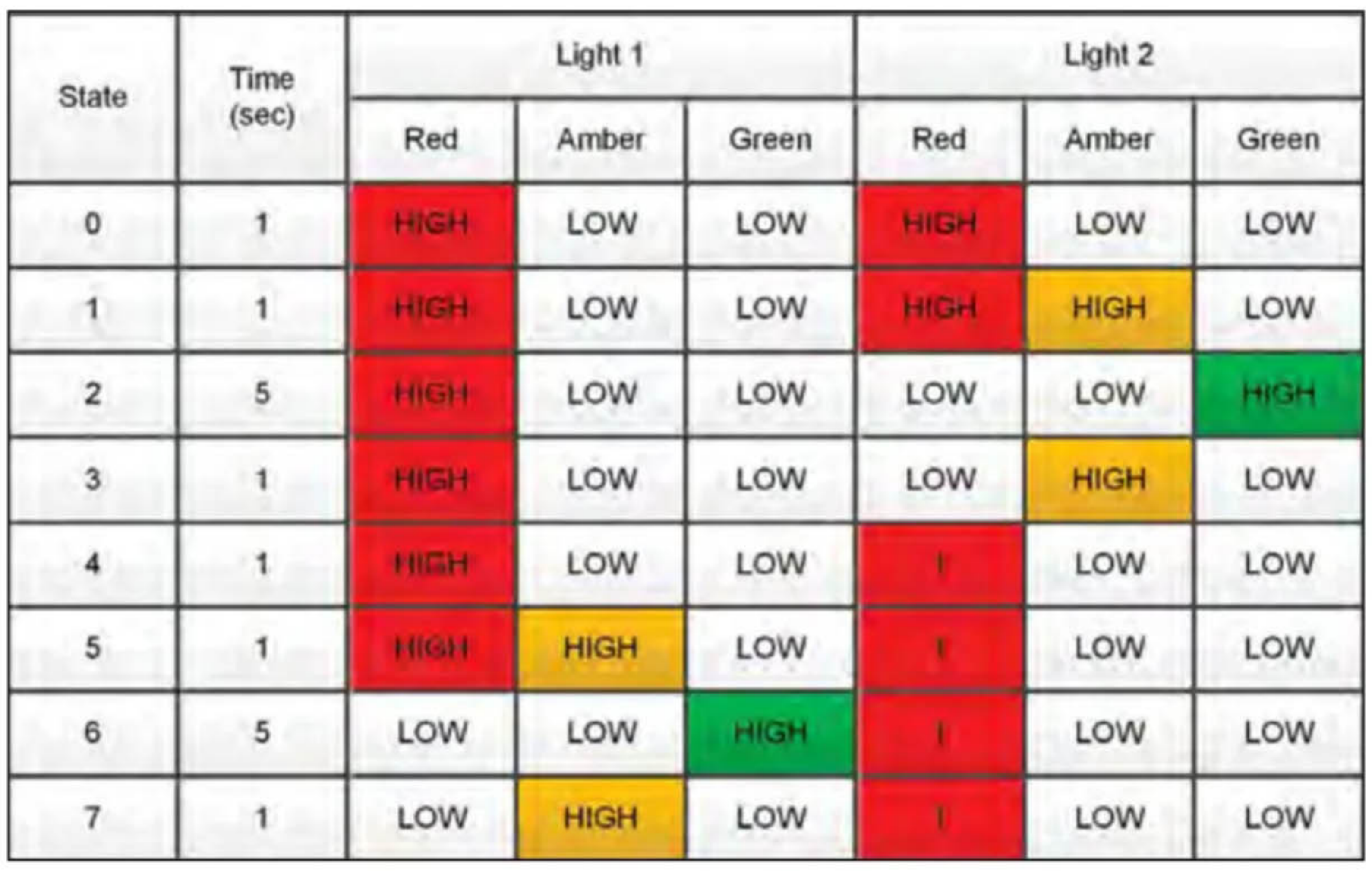
When the code is executed the two traffic lights should follow the sequence shown in the state table (Fig.2.10). Notice how States 2 and 6 are each to be displayed for five seconds while the remaining states are displayed for 1 second. These display times are achieved by introducing delays of appropriate length in the code. Note that it is only necessary to modify one line of code to change the timing of the lights. The code is very straightforward and should need no further explanation.

Listing 2.3 Code for the ESP32 two-way traffic light controller

```
/* ESP32 traffic lights. Outputs
red, amber and
green control signals for a 2-way
road junction
*/

// Define GPIO pins
const int Red1 = 12;
const int Amber1 = 13;
const int Green1 = 14;
const int Red2 = 18;
const int Amber2 = 19;
const int Green2 = 21;
const int delaytime = 1000;

void setup() {
// All pins configured as outputs
   pinMode(Red1, OUTPUT);
   pinMode(Amber1, OUTPUT);
   pinMode(Green1, OUTPUT);
   pinMode(Red2, OUTPUT);
   pinMode(Amber2, OUTPUT);
   pinMode(Green2, OUTPUT);
}

void loop() {
// State 0
   digitalWrite(Red1, HIGH);
   digitalWrite(Amber1, LOW);
   digitalWrite(Green1, LOW);
   digitalWrite(Red2, HIGH);
   digitalWrite(Amber2, LOW);
   digitalWrite(Green2, LOW);
   delay(delaytime);
   // State 1
   digitalWrite(Red1, HIGH);
   digitalWrite(Amber1, LOW);
   digitalWrite(Green1, LOW);
   digitalWrite(Red2, HIGH);
   digitalWrite(Amber2, HIGH);
   digitalWrite(Green2, LOW);
   delay(delaytime);
   // State 2
   digitalWrite(Red1,  HIGH);
   digitalWrite(Amber1, LOW);
   digitalWrite(Green1, LOW);
   digitalWrite(Red2, LOW);
   digitalWrite(Amber2, LOW);
   digitalWrite(Green2, HIGH);
   delay(5 * delaytime);
   // State 3
   digitalWrite(Red1, HIGH);
   digitalWrite(Amber1, LOW);
   digitalWrite(Green1, LOW);
   digitalWrite(Red2, LOW);
   digitalWrite(Amber2, HIGH);
   digitalWrite(Green2, LOW);
   delay(delaytime);
   // State 4
   digitalWrite(Red1, HIGH);
   digitalWrite(Amber1, LOW);
   digitalWrite(Green1, LOW);
   digitalWrite(Red2, HIGH);
   digitalWrite(Amber2, LOW);
   digitalWrite(Green2, LOW);
   delay(delaytime);
   // State 5
   digitalWrite(Red1, HIGH);
   digitalWrite(Amber1, HIGH);
   digitalWrite(Green1, LOW);
   digitalWrite(Red2, HIGH);
   digitalWrite(Amber2, LOW);
   digitalWrite(Green2, LOW);
   delay(delaytime);
   // State 6
   digitalWrite(Red1, LOW);
   digitalWrite(Amber1, LOW);
   digitalWrite(Green1, HIGH);
   digitalWrite(Red2, HIGH);
   digitalWrite(Amber2, LOW);
   digitalWrite(Green2, LOW);
   delay(5 * delaytime);
   // State 7
   digitalWrite(Red1, LOW);
   digitalWrite(Amber1, HIGH);
   digitalWrite(Green1, LOW);
   digitalWrite(Red2, HIGH);
   digitalWrite(Amber2, LOW);
   digitalWrite(Green2, LOW);
   delay(delaytime);
}
```

Configuring digital inputs

There are numerous applications in which you might need to make use of one or more of the ESP32's digital inputs. Such applications include sensing the state of buttons and switches as well as a host of sensors with digital outputs, such as float switches, pressure switches, touch switches, microswitches, proximity switches and magnetic reed switches. These devices can all make use of a straightforward connection to a GPIO



State	Time (sec)	Light 1			Light 2		
		Red	Amber	Green	Red	Amber	Green
0	1	HIGH	LOW	LOW	HIGH	LOW	LOW
1	1	HIGH	LOW	LOW	HIGH	HIGH	LOW
2	5	HIGH	LOW	LOW	LOW	LOW	HIGH
3	1	HIGH	LOW	LOW	LOW	HIGH	LOW
4	1	HIGH	LOW	LOW	I	LOW	LOW
5	1	HIGH	HIGH	LOW	I	LOW	LOW
6	5	LOW	LOW	HIGH	I	LOW	LOW
7	1	LOW	HIGH	LOW	I	LOW	LOW

Fig.2.10. Sequence table for the traffic lights controller.

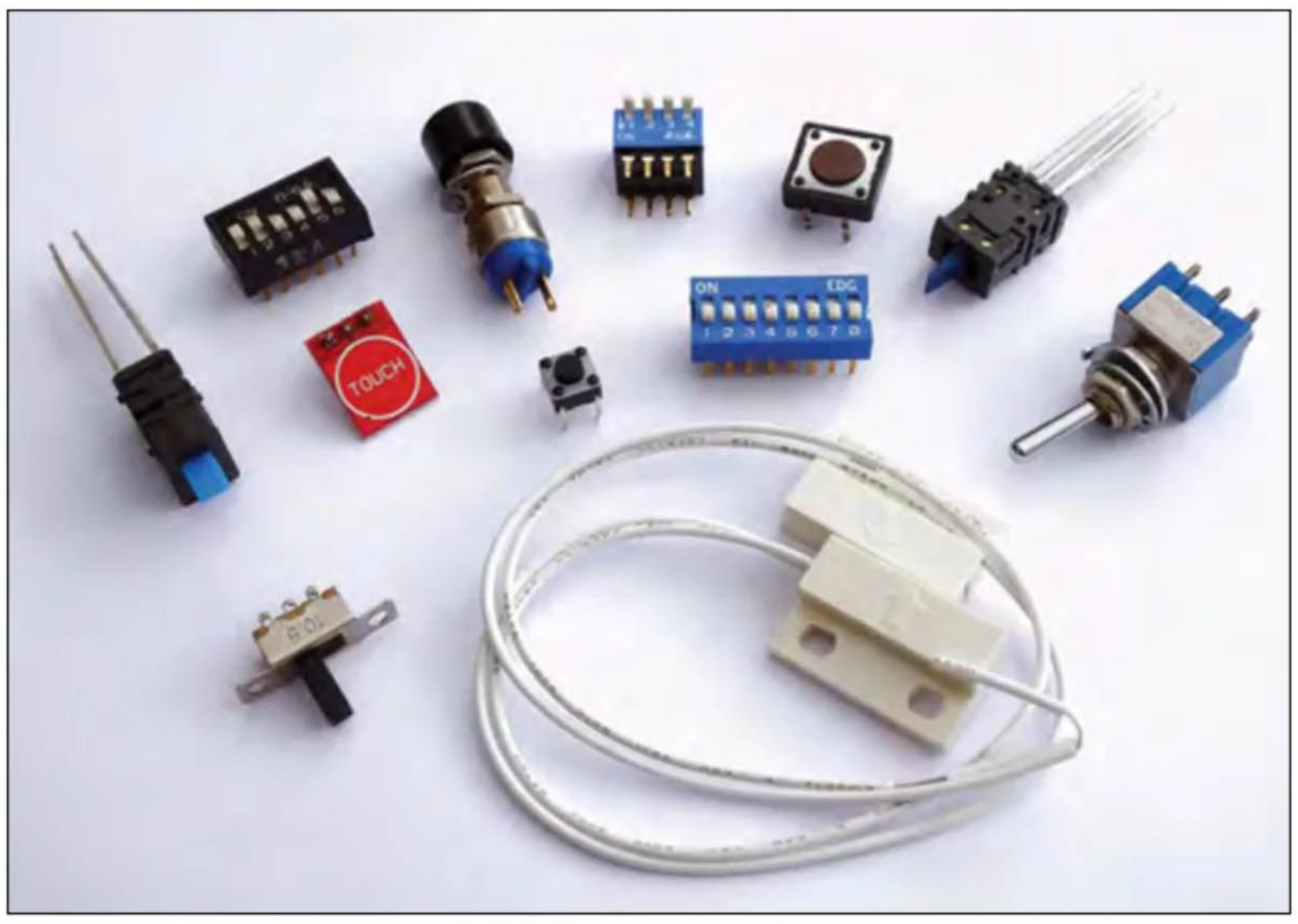


Fig.2.11. A selection of commonly available buttons and switches ideal for use with the ESP32.

Gotcha!

GPIO pins 34, 35, 36 and 39 are only available for digital input and do not have internal pull-up and pull-down resistors. So, if you need to use these pins for digital inputs, remember to fit external pull-up or pull-down resistors.

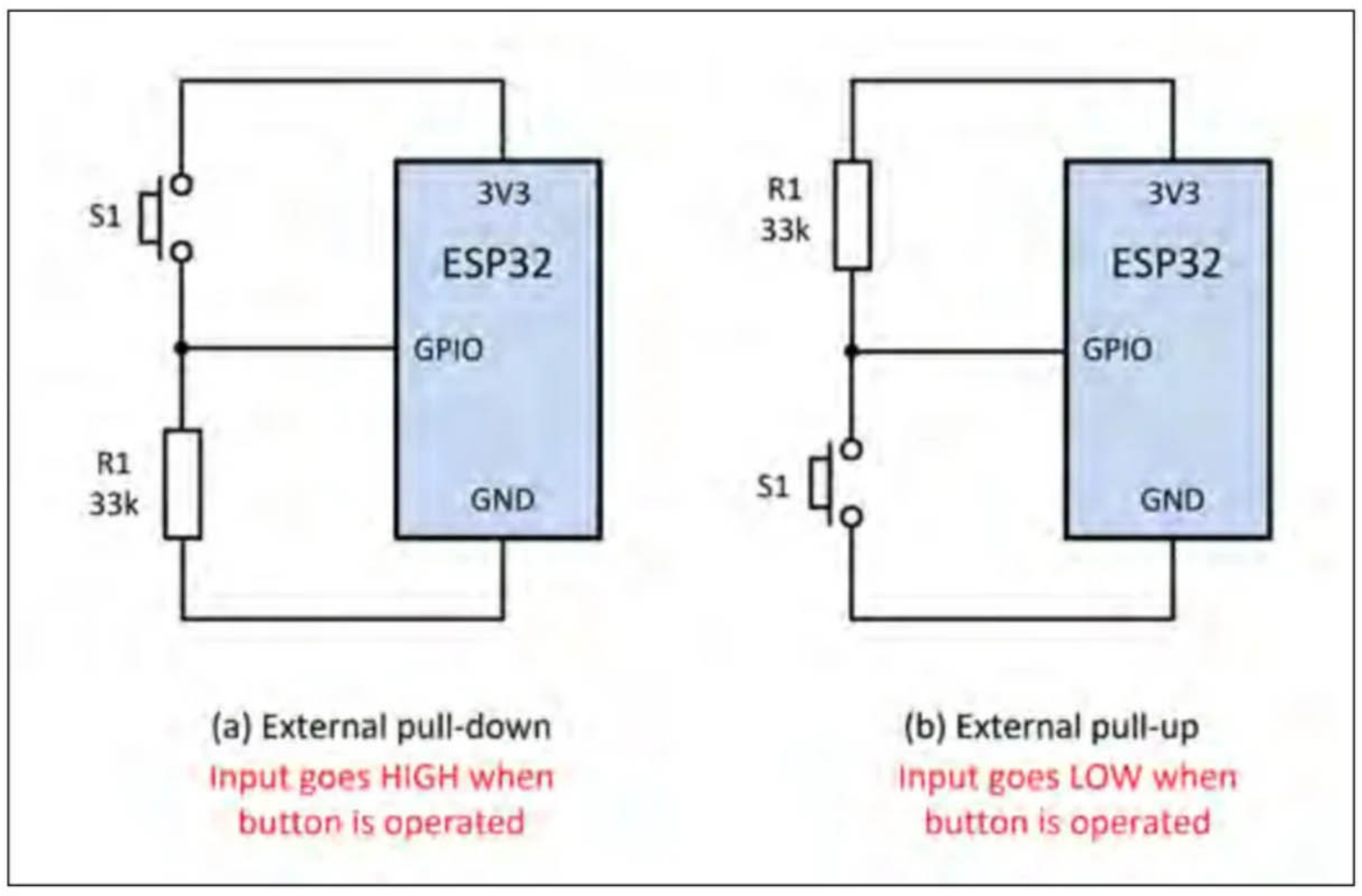
input pin. A selection of commonly available buttons and switches is shown in Fig.2.11.

Pull-up or pull-down?

We need our digital inputs to be in one of two states, either HIGH or LOW with nothing in between. The former is usually defined as a voltage level that's greater than 75% of the nominal 3.3V supply, while the latter is taken as less than 25% of the same value (these two states correspond to more than 2.5V or less than 0.8V, respectively). Note that, by default, the GPIO inputs adopt a high-impedance state, and this invariably results in an unpredictable and indeterminate input Because of this you will often find that the GPIO pins are either pulled HIGH (pulled up) or pulled LOW (pulled down) by means of external resistors. Values are generally uncritical and can often be in the range 4.7kΩ to 100kΩ. Fig.2.12 illustrates the use of this technique.

Gotcha!

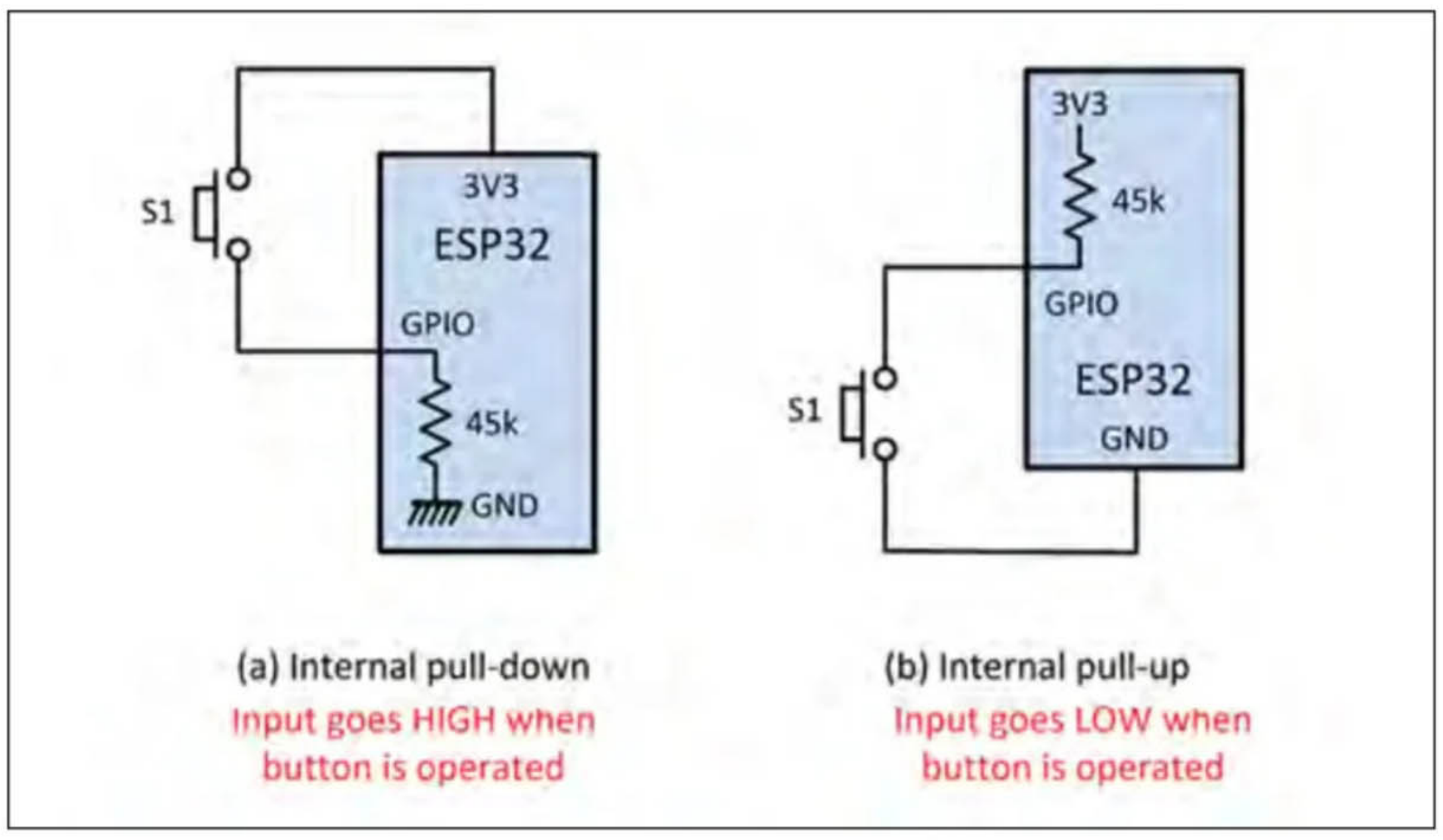
A problem arises with some types of mechanically actuated switch in which the switching action is not clean, and the output state fluctuates rapidly before settling to a steady value. This condition is referred to as 'switch bounce' (see Fig.2.14) and it can be undesirable in many applications. The effects of switch bounce can be reduced or eliminated by various means including the introduction of software delays or by slowing the switching action using a simple capacitor-resistor (C-R) network between the switch contacts and a GPIO input pin, as shown in Fig.2.15.


S1
3V3
ESP32
GPIO
R1
33k
GND
R1
33k
3V3
ESP32
GPIO
S1
GND

(a) External pull-down
Input goes HIGH when button is operated

(b) External pull-up
Input goes LOW when button is operated

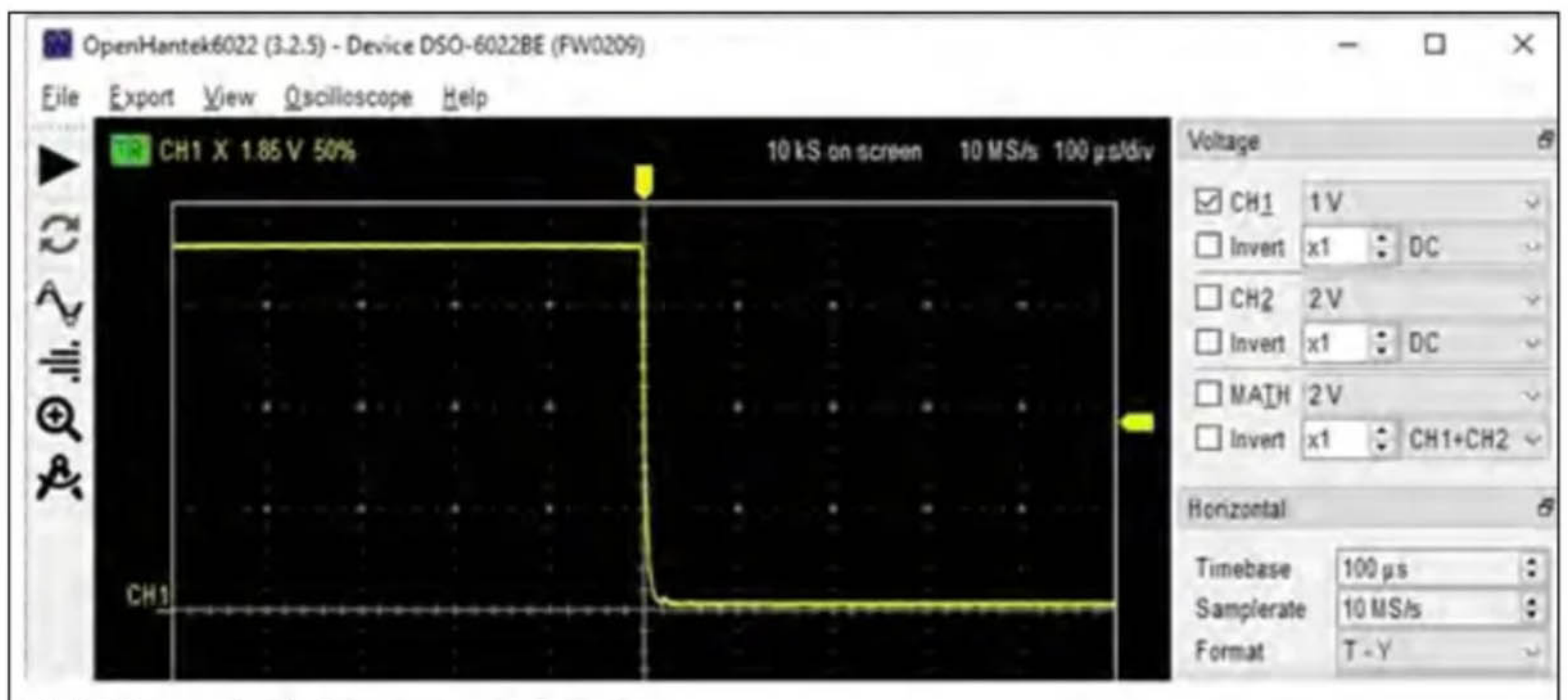
Fig.2.12. External pull-up and pull-down resistors.


S1
3V3
ESP32
GPIO
45k
GND
3V3
45k
GPIO
ESP32
S1
GND

(a) Internal pull-down
Input goes HIGH when button is operated

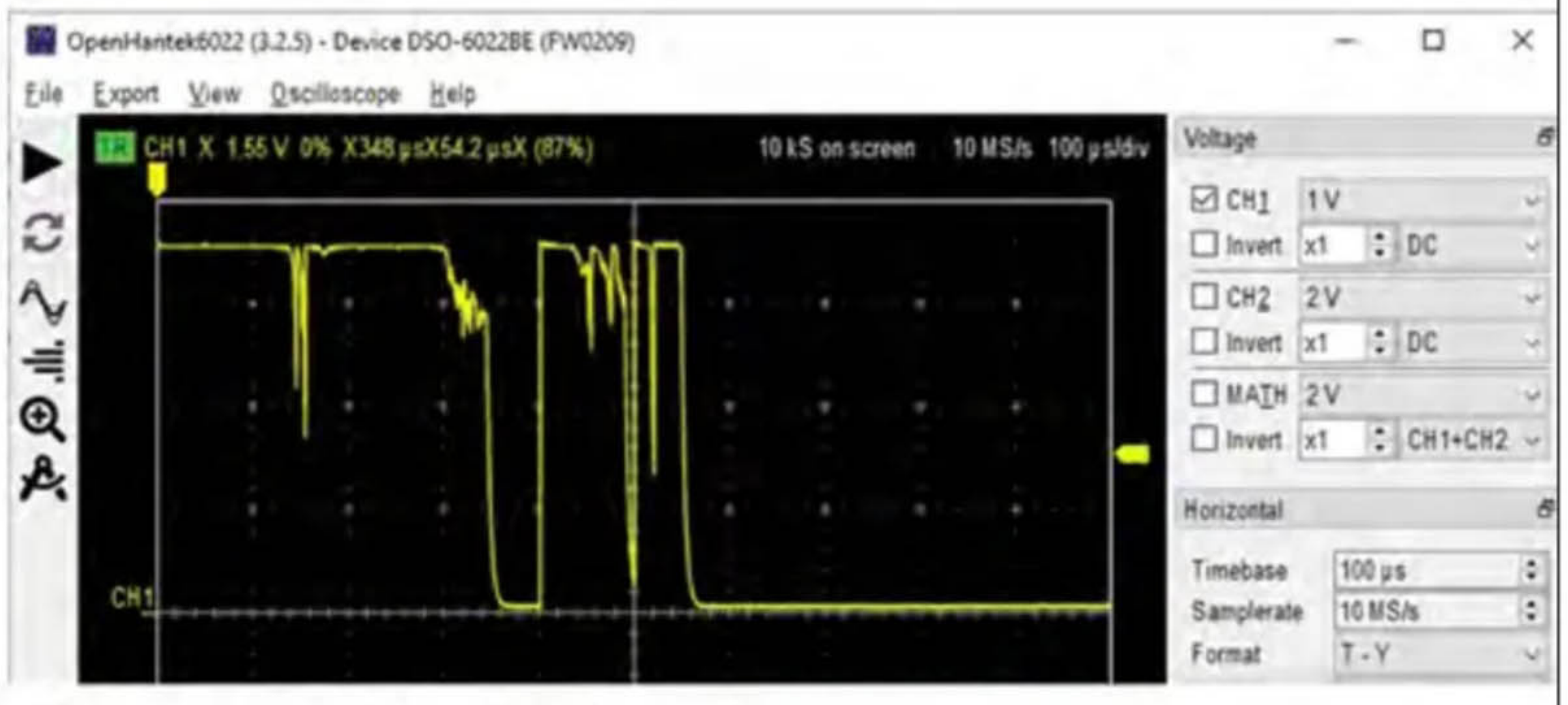
(b) Internal pull-up
Input goes LOW when button is operated

Fig.2.13. Equivalent internal pull-up and pull-down resistors.


OpenHantek6022 (3.2.5) - Device DSO-6022BE (FW0209)
File
Export
View
Oscilloscope
Help
TR
CH1 X 1.85 V 50%
10 kS on screen
10 MS/s
100 µs/div
CH1
Voltage
Horizontal

☑ CH1 1 V
☐ Invert x1 DC
☐ CH2 2 V
☐ Invert x1 DC
☐ MATH 2 V
☐ Invert x1 CH1+CH2
Timebase 100 µs
Samplerate 10 MS/s
Format T - Y

(a) No switch bounce evident


OpenHantek6022 (3.2.5) - Device DSO-6022BE (FW0209)
File
Export
View
Oscilloscope
Help
TR
CH1 X 1.55 V 0% X348 µsX54.2 µsX (87%)
10 kS on screen
10 MS/s
100 µs/div
CH1
Voltage
Horizontal

☑ CH1 1 V
☐ Invert x1 DC
☐ CH2 2 V
☐ Invert x1 DC
☐ MATH 2 V
☐ Invert x1 CH1+CH2
Timebase 100 µs
Samplerate 10 MS/s
Format T - Y

(b) Severe switch bounce present

Fig.2.14. Waveform showing switch bounce.

It's important to be aware that the ESP32 has a rather neat feature that allows inputs to be pulled HIGH or LOW *without* the need for external resistors. This can be achieved during GPIO initialisation using the `pinMode()` command, as shown in the following fragment of code:

```
pinMode(18, INPUT); D18 is a default digital input
pinMode(19, INPUT_PULLUP); D19 is a digital input
pulled high
pinMode(21, INPUT_PULLDOWN); D21 is a digital
input pulled low
```

The effect of `INPUT_PULLDOWN` and `INPUT_PULLUP` is illustrated in Fig.2.13. Note that the ESP32's internal pull-up/pull-down resistors have a nominal value of 45kΩ.

Practical project (Part 1)

Our *Practical Project* involves the design, construction and coding for a simple motor controller. The minimal circuit and wiring layout for our prototype motor controller are shown in Fig.2.16 to Fig.2.18 (the pin connections for the LED were shown in Fig.1.21 last month). We've used 4.7kΩ external pull-down resistors in conjunction with the RUN and STOP switches and a 330Ω series resistor for the LED (later, we will be replacing this with a small motor and driver).

Having connected the circuit, made the USB connection to your PC and started the IDE, you can enter (or download) the code shown in Listing 2.4. When you execute the code, you should see that the LED becomes illuminated when the RUN button is pressed. The LED should then remain illuminated until the STOP button is operated. A subsequent press of the RUN button should restart the LED. Now let's walk through the code.

The first few lines of code define the constants used to allocate the GPIO pin numbers to the two buttons and LED. Next, we define the variables that will represent the states of the two buttons and the

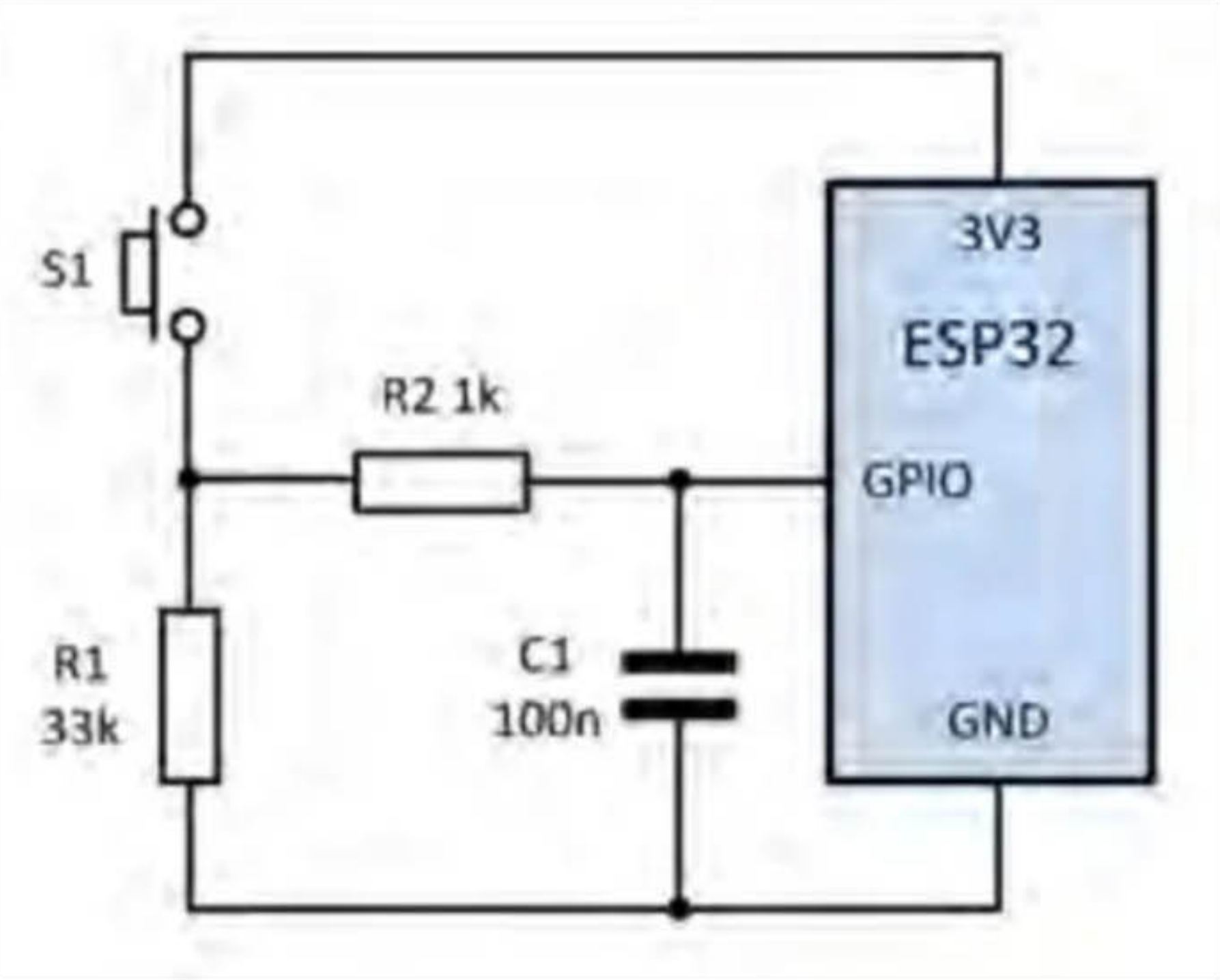

S1
R2 1k
3V3
ESP32
GPIO
R1
33k
C1
100n
GND

Fig.2.15. Using a C-R network to reduce the effects of switch bounce.

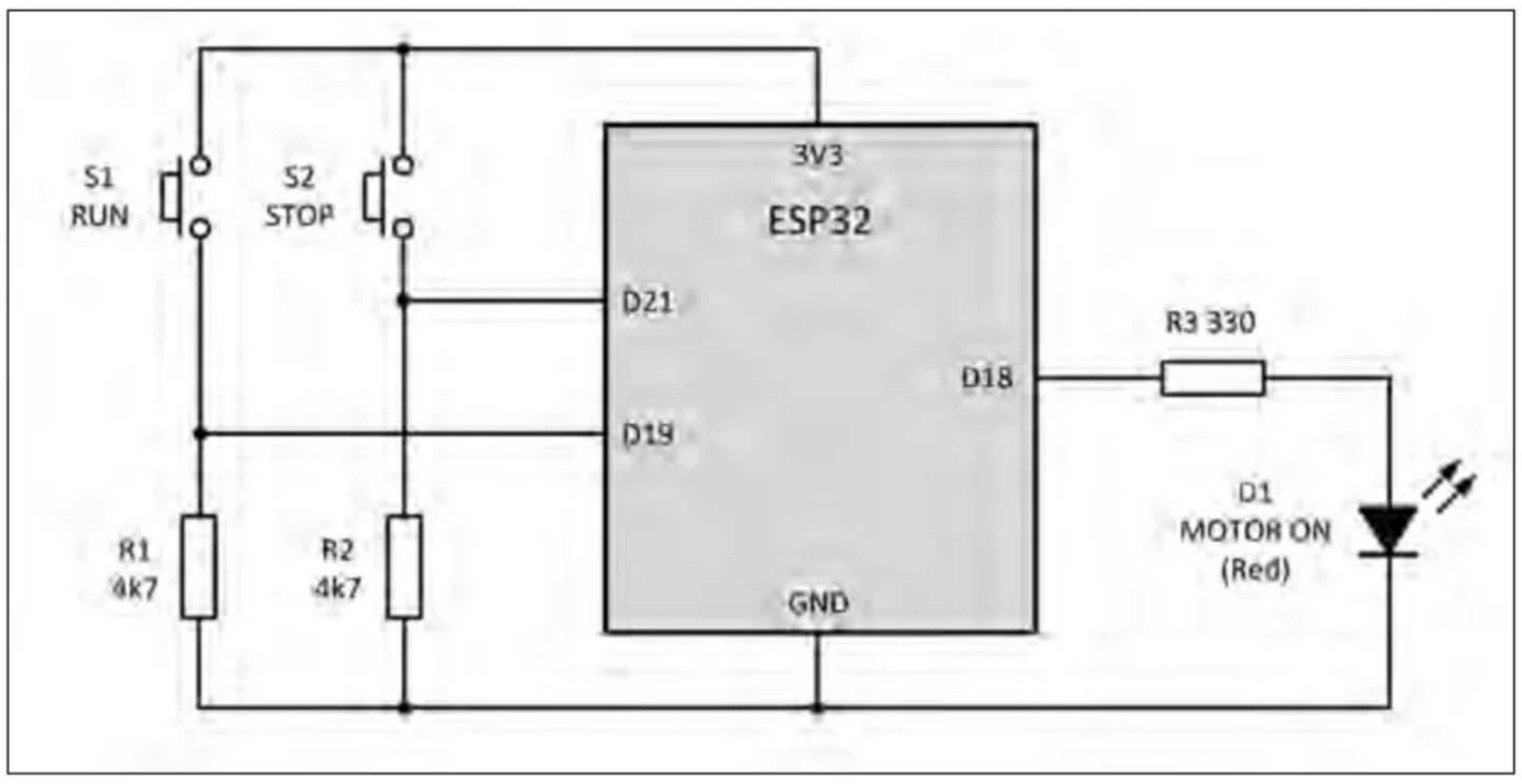


Fig.2.16. Circuit for the prototype motor controller.

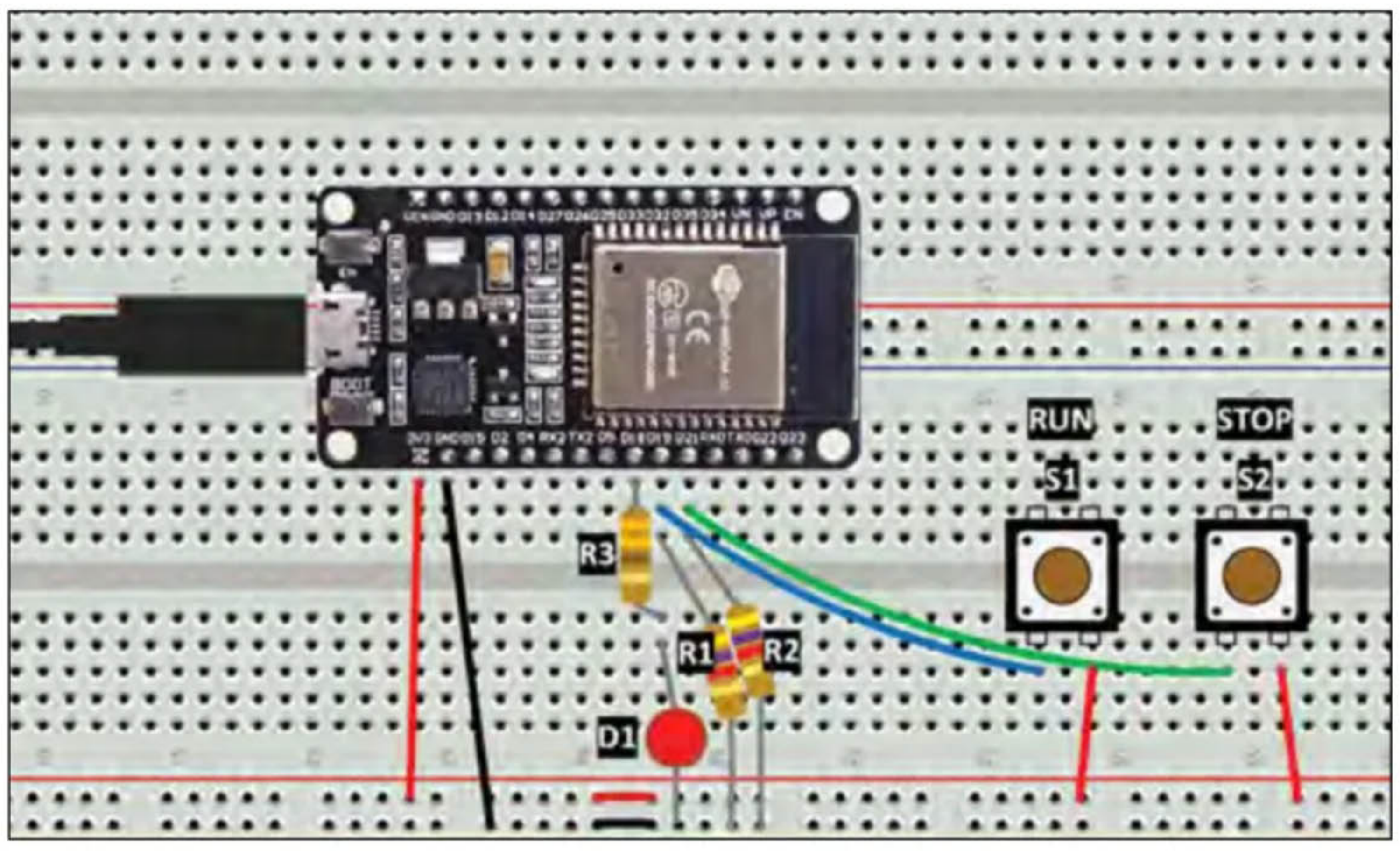


Fig.2.17. Circuit layout for the prototype motor controller (based on a 30-pin ESP32 board).

LED output. Finally, the GPIO pins are then configured in the `setup()` code.

The main loop of the code repeats forever and starts by reading the state of the two buttons. Having read the button states we decide what to do depending upon the current state of each button. If the RUN button has been operated, we activate the LED but if the STOP button has been operated, we turn the LED off. Later, in *Coding workshop* we will take a close look at how these decisions are made.

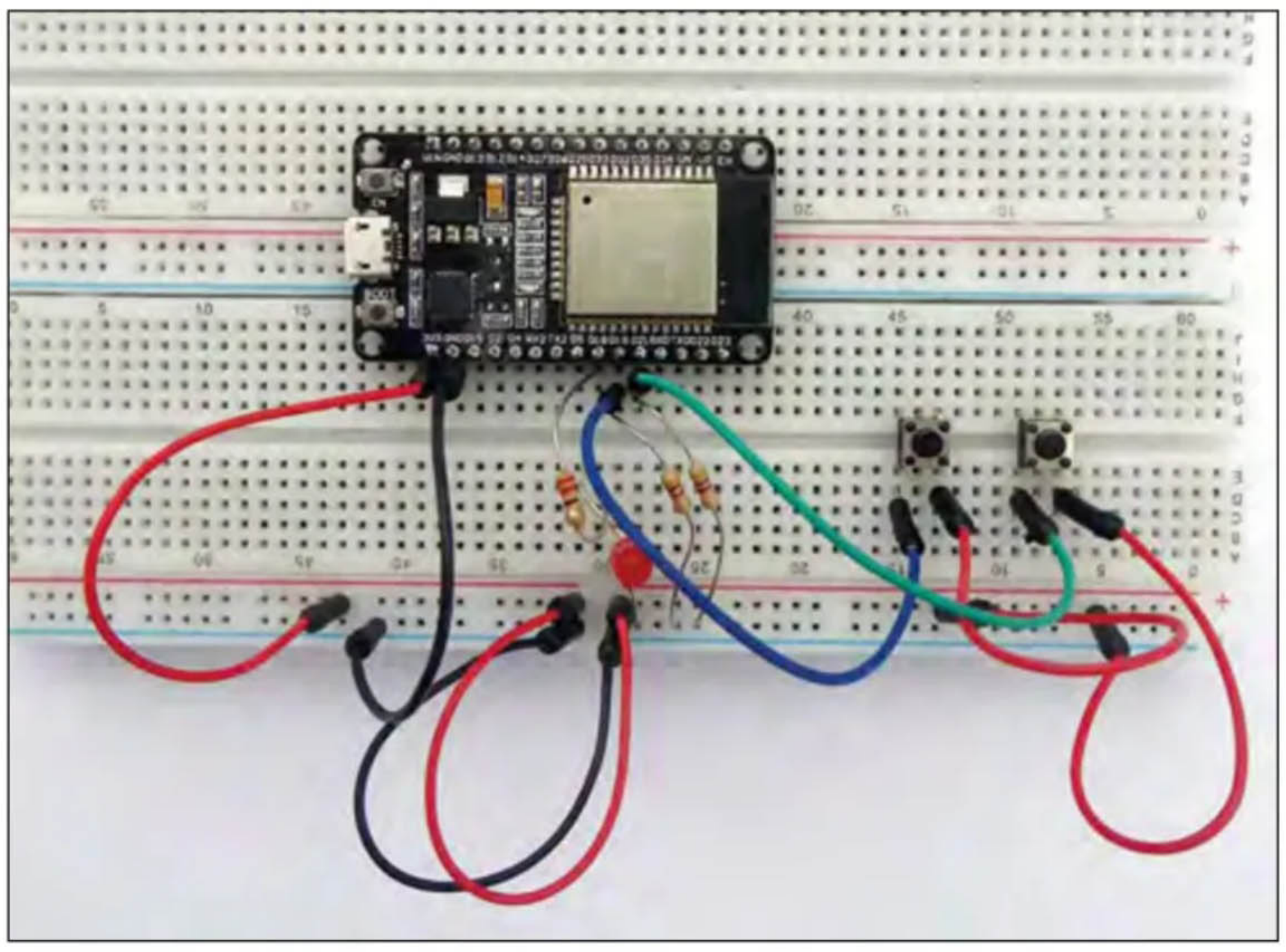


Fig.2.18. Wiring for the prototype motor controller (based on a 30-pin ESP32 board).

Fig.2.19. Solid-state relays designed for controlling AC mains with maximum load currents ranging from 10A to 45A.

Having verified that the prototype works we can add a small motor to be controlled by the RUN and STOP buttons. Bearing in mind the current limitation mentioned earlier we will need to add some extra circuitry to supply the greater current required. We will return to this in *Part 2* a little later, but first it's worth considering how to drive loads that demand more current or voltage than the ESP32 can provide.

Driving high-current/high-voltage loads

The ESP32's current drive capability is sufficient for a lot of purposes (including illuminating an LED) but not enough to operate actuators, motors, lamps and many other 'real-world' output devices. Fortunately, the problem of driving the vast majority of high-voltage and high-current loads can be easily solved using one or more miniature relays. These electromechanical devices comprise a coil wound on a high-permeability core and a moving armature mechanically linked to a set of contacts that make and break when the device is actuated. When sufficient current is applied to the coil of the relay the resulting magnetic field will cause the soft iron armature to pull-in and this in turn will open or close the relay's electrical contacts. A typical miniature PCB mounted relay will operate from a 5V supply, and its contacts will pull-in at around 75% of this value. The specifications of such a relay are listed in Table 2.1. Solid-state relays (like those shown in

Listing 2.4 Simple motor control

```
/* ESP32 button control example. One button will be
for Start and the other for Stop. The output will be
indicated on an LED*/

// Define constants for the GPIO pins
const int runButton = 19;  // GPIO19
const int stopButton = 21; // GPIO21
const int ledOutput = 18;  // GPIO18

// Define variable states
int runButtonState = LOW;    // No user input yet so both
int stopButtonState = LOW;   // buttons will be low
int outputState = LOW;       // Initially this must be low

void setup() {
   // Initialize the LED pin as an output
   pinMode(ledOutput, OUTPUT);
   // Initialize the two push button pins as an inputs
   pinMode(runButton, INPUT);
   pinMode(stopButton, INPUT);
}

void loop() {
   // Read the current state of the two buttons
   runButtonState = digitalRead(runButton);
   stopButtonState = digitalRead(stopButton);
   // Check if the buttons are pressed
   if (runButtonState == HIGH) {
     outputState = HIGH;
     digitalWrite(ledOutput, HIGH);
   }
   if (stopButtonState == HIGH) {
     outputState = LOW;
     digitalWrite(ledOutput, LOW);
   }
}
```

Table 2.1 Electrical specifications of a typical miniature relay

Parameter	Value
Nominal operating voltage	5V DC
Nominal operating current	73mA
Maximum load rating	AC 250V 10A, DC 30V 10A
Pull-in voltage (typical)	3.8V
DC coil resistance	70Ω
Power consumption (typ)	0.36W
Operating time (max)	10ms
Release time (max)	5ms
Contact resistance (max)	0.11Ω
Operating life	100,000 operations at rated load
Maximum switching rate	30 operations per second

Fig.2.19) can be used to control much greater AC mains loads of up to 45A, but can still operate from a low voltage input.

It is important to note from Table 2.1 that the relay coil requires an operating current that's well beyond the output drive capability of the ESP32. We therefore need an interface that will provide the extra current required. Fortunately, this can be little more than a low-power transistor and a handful of other components, as we shall see later in *Part 2* of our *Practical Project*.

A neater, and often more cost-effective alternative is that of making use of a ready built relay interface board. Luckily, there are quite a few to choose from and the most common types are fitted with two, four or eight relays with each relay having its own driver circuit.

Fig.2.20 shows a typical four-channel relay board. The board has a transistor driver and an opto-isolator for each channel. Similar boards can be purchased very cheaply (often less than £5) and so it is invariably more cost effective to purchase one of these boards rather than attempt to build one yourself.

Coding relay outputs

It's very easy to control one or more relays using just a few lines of simple code. First, you will need to make sure that you define the digital output pins to which the relays are connected using a line of the form:

```
int pump = 18;     // Pump connected via a relay
                   on GPIO pin-18
int heater = 21;   // Heater connected via a
                   relay on GPIO pin-21
```

Next you will need to add a couple of lines into the `setup()` code block, as follows:

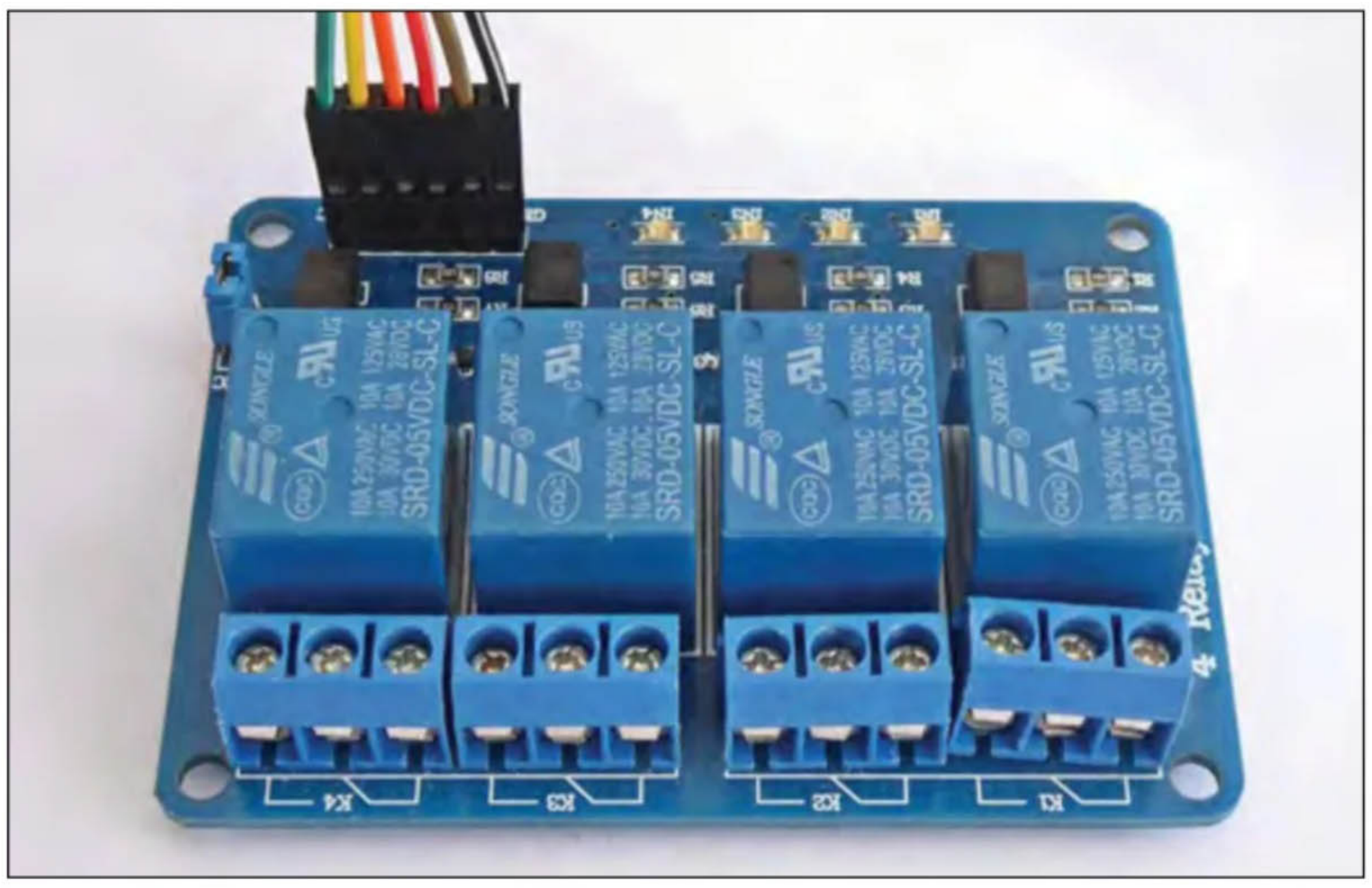


Fig.2.20. A four-channel relay interface board.

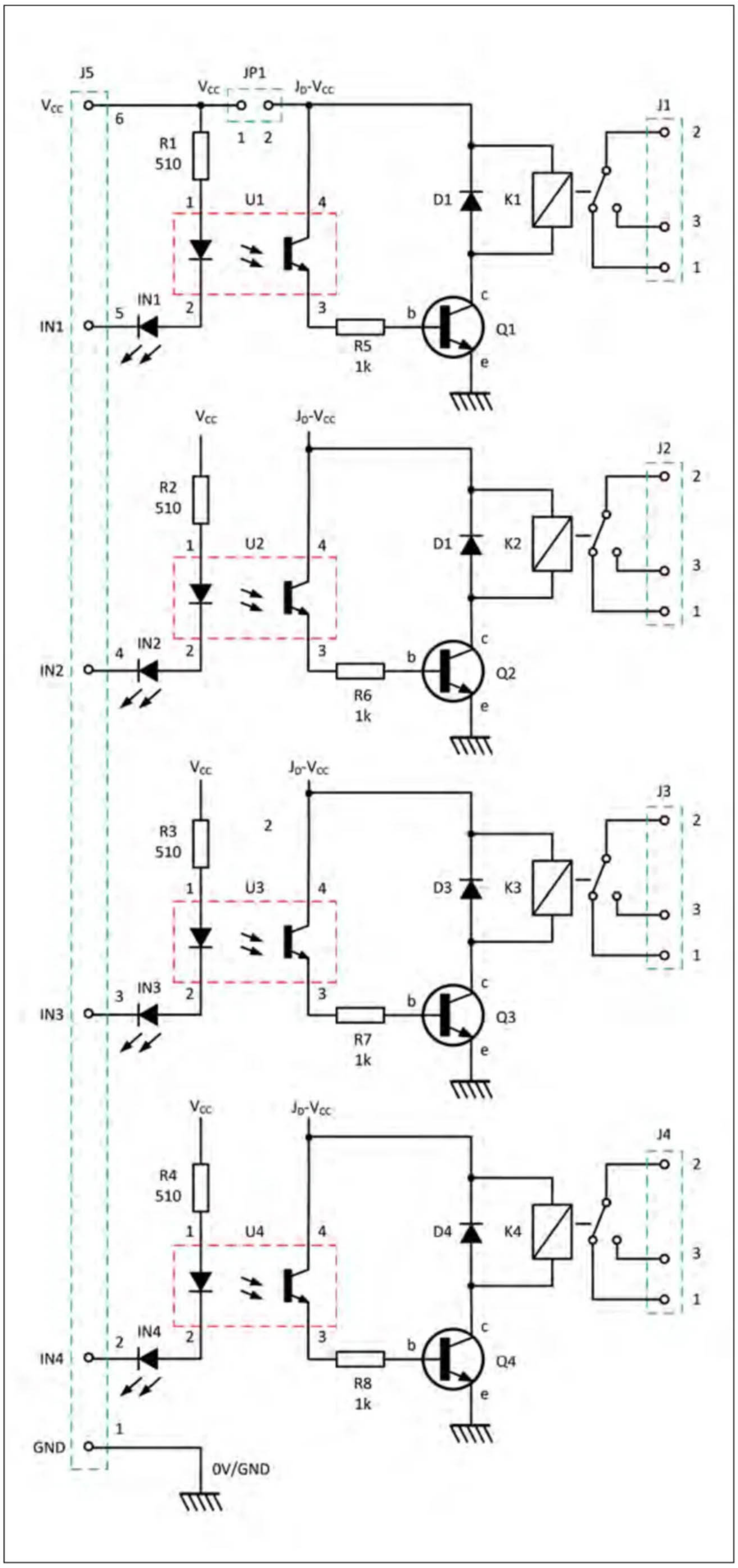


Fig.2.21. Circuit of the four-channel relay interface board.

```
pinMode(pump, OUTPUT);     // Pump is configured
                           as an output
pinMode(heater, OUTPUT);  // Heater is
                           configured as an output
```

The relays and their respective loads can be turned on and off incorporating the following lines of code at appropriate points in the main program loop:

```
digitalWrite(pump, LOW);    // Turn the pump off
digitalWrite(pump, HIGH);   // Turn the pump on
digitalWrite(heater, LOW);  // Turn the heater off
digitalWrite(heater, HIGH); // Turn the pump on
```

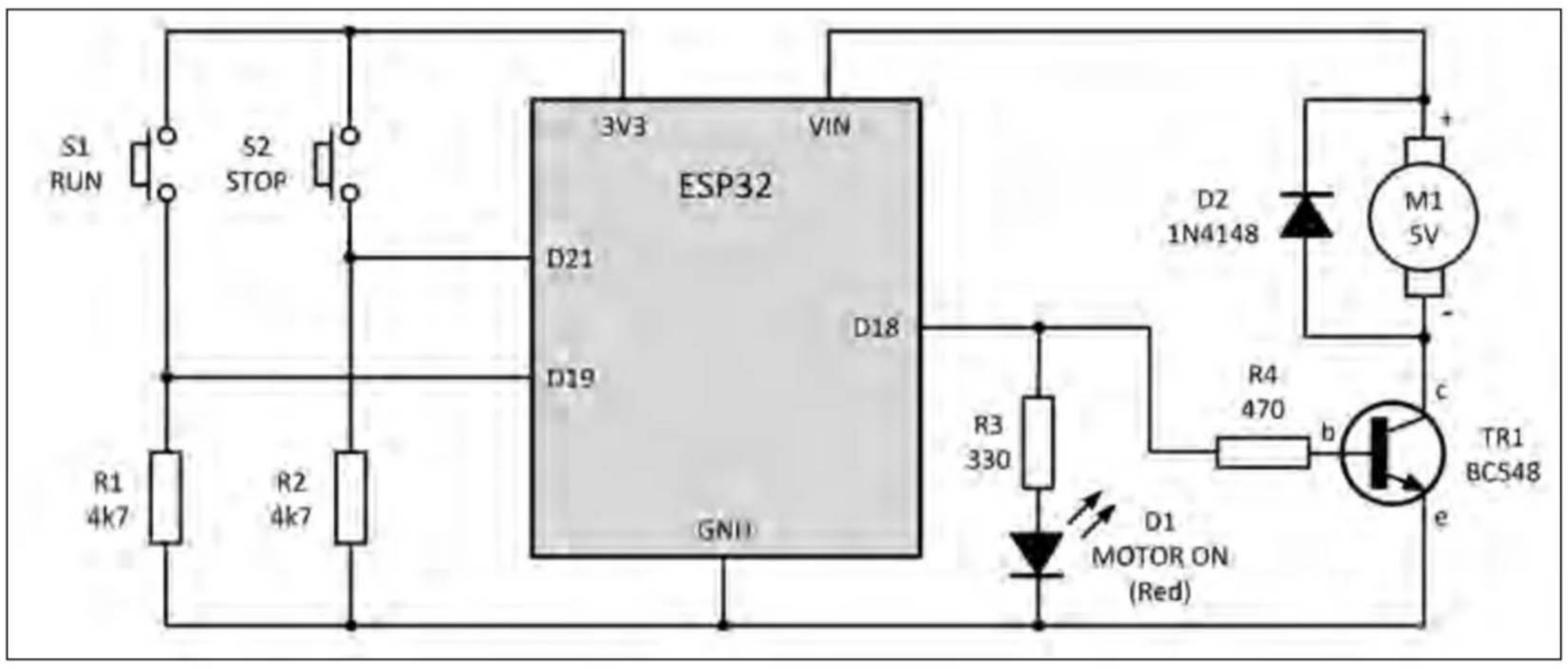


Fig.2.22. Adding a transistor driver and motor to the output of the motor controller.

Practical project (Part 2)

Now let's return to the *Practical project*. We can add a motor to the output of the ESP32 (GPIO pin-2) using a transistor driver similar to those used in the four-channel relay interface shown in Fig.2.20. The modified circuit and wiring diagrams for the motor controller are shown in Figs.2.22 and 2.23 respectively. The motor is a low-cost miniature 5V component fitted with an integral gearbox. TR1 provides the larger current drive for the motor (typically 50 to 100mA) and acts as a switch which is open when the GPIO pin-2 goes LOW and is closed when GPIO pin-2 goes HIGH (see Fig.2.24). Diode D1 is used to protect TR1 from the back EMF generated when the motor current flow ceases. Similar diodes are used whenever inductive loads (such as relays, solenoids and motors) are used as loads.

Coding workshop

Being able to make decisions based on what's happening and then acting on this information in different ways is an essential pre-requisite of any programming language. The C-code used in the Arduino IDE provides you with a variety of different conditional constructs that allow you to do this. Simple decisions can be made using nothing more than `if` and `else`, while loops can be controlled using `while`, `do while`, `for` and `loop`. We will look at all of these, starting with `if` and `else`.

The `if` construct

The `if` construct is the simplest of all the condition constructs. It is used when a statement (or a series of statements) should be executed when a particular condition prevails. The basic syntax is as follows:

```
if (conditional expression) {
  // code to be executed if true,
  // each statement ending with ;
}
```

An example might be counting items into batches of 10 on a conveyor. Let's assume that we need to operate an LED when the count reaches (or exceeds) ten items. The following fragment of code would do this:

```
if (count >=10) {
  digitalWrite(fullLED, HIGH);
}
```

If the value of count is less than 10 then the condition evaluates false and the statement following the condition is simply ignored. However, if the value of count is 10 or greater then the condition evaluates true and the statement following the condition is executed. In some applications it can be appropriate to use a series of `if` statements to detect various conditions and to act on them accordingly.

The `if ... else` construct

The `if ... else` construct is very straightforward. The basic syntax is just:

```
if (conditional expression) {
  // code to be executed if true,
  // each statement ending with ;
} else {
  // code to be executed if false
  // each statement ending with ;
}
```

Here's a simple example. Let's assume that we are monitoring an analogue voltage and wish to set a threshold of 512 as the value at which a green LED should become illuminated and, below this value, we would like a red LED to be turned on. Our `if … else` construct would then look something like this:

```
if (inVoltage >=128) {
  digitalWrite(greenLED, HIGH);
} else {
  digitalWrite(redLED, HIGH);
}
```

Unfortunately, this isn't quite the whole story. The red and green status LEDs should be mutually exclusive and so we might need to ensure that when one LED is turned on the other LED is turned off. There are various ways that we could do this. We could either set them both off before we arrive at the `if else` construct or we could turn one on and the other off

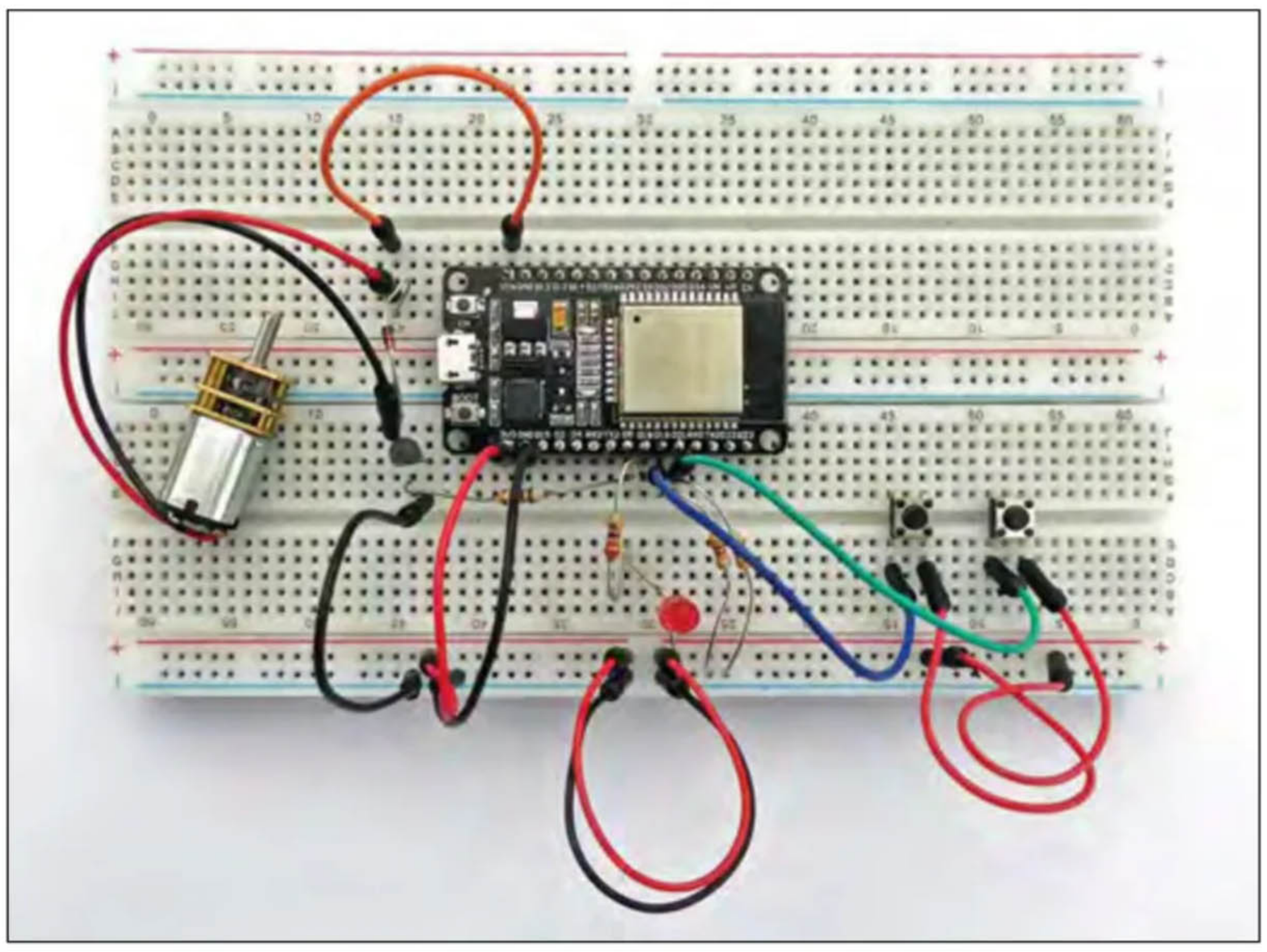
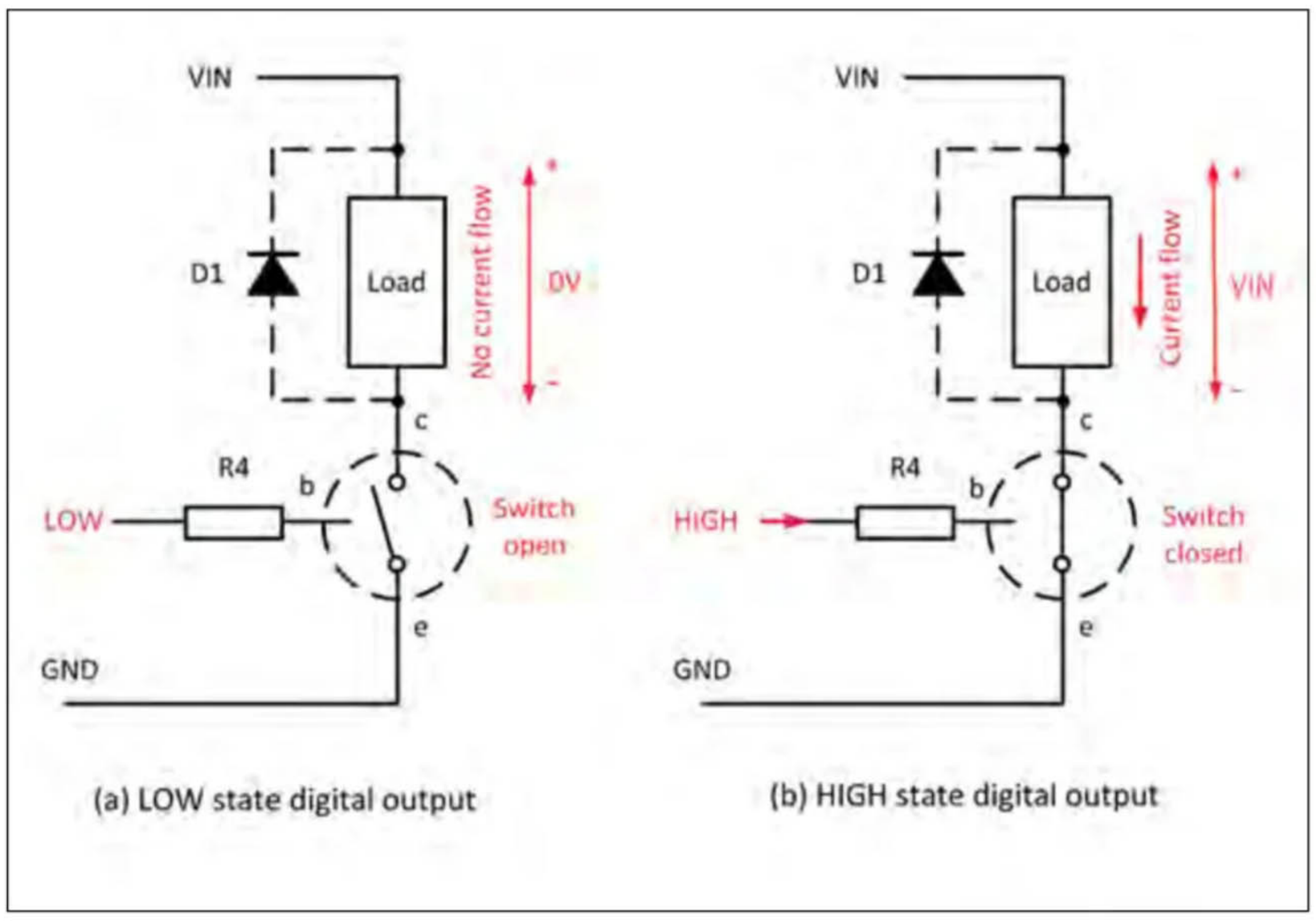


Fig.2.23. Wiring of the modified motor controller.



(a) LOW state digital output

(b) HIGH state digital output

Fig.2.24. Switching action of the transistor in Fig.2.22.

within a construct containing more than one statement (ie, a *compound* construct).

Using the first method we might have:

```
// start with both LEDs off
digitalWrite(redLED, LOW);
digitalWrite(greenLED, LOW);
// now decide which LED to put on
if (inVoltage >=512) {
  digitalWrite(greenLED, HIGH);
} else {
  digitalWrite(redLED, HIGH);
}
```

The other possibility is:

```
// Read the input voltage and
// put the red or green LED on
if (inVoltage >=512) {
  digitalWrite(greenLED, HIGH);
  digitalWrite(redLED, LOW);
} else {
  digitalWrite(redLED, HIGH);
  digitalWrite(greenLED, LOW);
}
```

Notice how in this example we've compounded several statements after the `if` and `else`. The curly braces, `{` and `}`, help make the logic clear and unambiguous. Of course, at some point earlier in the code we would have to define the pins that we are using to control the two LEDs and initialise the variables (`inVoltage`, `redLED` and `greenLED`).

Conditions

There are several logical conditions that we can test for, and they are summarised in Table 2.2. Notice how an exclamation mark (`!`) is used to denote 'not'.

In the previous example you should have noticed the `>=` condition that we used to find out whether the input voltage has exceeded the threshold value of 512. The '*greater than or equal to*' condition isn't the only one that we have to play with, as Table 2.2 shows.

The `while` construct

The `while` construct provides you with a means of continuously executing one or more statements until a condition evaluates false. The loop containing the statement (or statements) will continue to be executed as long as the condition remains true but, as soon as it becomes false the loop will terminate and execution will continue with the next subsequent statement. The basic syntax is:

```
while (conditional expression) {
  // statements to be executed if true,
  // each ending with ;
}
```

Here's an example that shows how a belt motor could be controlled using a `while` loop. The belt motor will run for as long as it takes for an item placed on the belt to reach a limit switch. Note that we must check the status of the limit switch inside the loop. If we forget to do this the motor will run forever!

```
while (limitSwitchStatus == LOW) {
  // Limit not reached so run the motor
  digitalWrite(motorRun, HIGH);
  // Check to see if anything has changed?
  limitSwitchStatus = digitalRead(limitSwitch);
}
```

By making the conditional expression dependent on the value of a counter modified inside the loop we have a simple means of performing one or more statements a predetermined number of times, as follows:

```
count = 0;
while (count < 50) {
  // code to be executed 50 times
  // each statement ending with ;
  count = count+1;
}
```

In this `wait` loop we increment the counter on every pass through the loop until it reaches 50, at which point the conditional expression evaluates to false and execution continues with the next statement in the code. Note that there's a neater way of incrementing the count value, as follows:

```
count = 0;
  while (count < 50) {
  // code to be executed 50 times
  // each statement ending with ;
  count++;
}
```

In this case, `count++` is used to 'post-increment' the value of `count`. In other words, it takes the current value of `count`, adds one to it and places the new value back into the `count` variable.

Finally, here's an example showing how a simple wait loop could be used to flash an alarm LED ten times:

```
flashCount = 0;
while(flashCount < 10) {
  digitalWrite(alarmLED, HIGH);
  delay(500); // wait half a second
  digitalWrite(alarmLED, LOW);
  delay(500); // wait half a second
  flashCount++;
}
```

The `do ... while` construct

The `do ... while` loop works in a similar fashion to the while loop but with the exception that the condition is tested at the end of the loop, not the beginning. This means that the statements within the loop will always be executed at least once. The syntax is as follows:

Table 2.2 Conditions

Code	Meaning	Notes
`a == b`	a is equal to b	True if a and b have the same value
`a != b`	a is not equal to b	True if a and b have the different values
`a > b`	a is greater than b	True if a is larger than b (but not true if they have the same value)
`a < b`	a is less than b	True if a is smaller than b (but not true if they have the same value)
`a >= b`	a is greater than or equal to b	True if a is larger than b (and also true if they have the same value)
`a <= b`	a is less than or equal to b	True if a is smaller than b (and also true if they have the same value)

```
do {
  // code to be executed at least once,
  // each statement ending with ;
} while (conditional expression);
```

Here’s an example of reading a pressure sensor and waiting a short time for the output to reach a steady value:

```
do {
  delay(100);
  // wait for the value to settle
  cp = readPressure();
  // read the pressure sensor
} while (x < 10);
```

In this example we are calling the `readPressure()` function 10 times before arriving at the final value returned from the sensor.

The `for` loop construct

The `for` loop is widely used in almost every computer language, and C is no exception. The construct is used to repeat a statement (or series of statements) whenever a condition evaluates true. If the condition evaluates false, then the loop is exited, and execution continues with the statement that immediately follows the loop. The `for` loop must be initialised at the outset and thus is a little more complex than the `while` loop. The basic syntax is:

```
for (loop initialization, conditional expression,
increment) {
  // statements to be executed if true,
  // each ending with ;
}
```

As with the `while` loop, a counter is often used to control the loop, and this is incremented or decremented each time round the loop. This makes the construct ideal for use in any repetitive application, for example, checking the status of several I/O lines. It is important to remember that loop initialisation occurs only once and before the loop is executed for the first time.

The next code fragment shows how the ASCII character set can be sent to the serial monitor. Note that, for this to run, we would first need to initialise the serial port interface using a line such as `Serial.begin(9600)`. Note also that we have declared the count variable (`i`) within the loop initialisation itself.

```
for (int i = 0; i <= 63; i++)
{
  testValue = 64 + i;
  Serial.print(testValue);
  Serial.print(“\n”);
  delay(100);
}
```

Program structure and layout

By now you should have gained some idea of what code looks like and how it is structured, but before we go any further it is well worth explaining the layout in a little more detail. You may have noticed that the first few lines of code in a program usually take the form of a heading enclosed between pairs of characters, `/*` and `*/`, which constitute a *comment block*. Everything between these two characters is taken as plain text and, since this has no effect on program execution you can use as many lines of text here as you want.

The code’s title comment block is usually followed by several variable declarations. The reason behind this is simply that, in the C language, variables *must always be declared* before they are used. In fact, declarations don’t have to be placed at the beginning of the program code but the point at which they are declared (ie, their position in the program) can impose restrictions on the scope over which they can be used. However, since we often need to use variables on a global basis (ie, anywhere in our program code) we will often place them *before* any of the other code. Declarations involve assigning a variable type (see last month), a name and (optionally) an initial value.

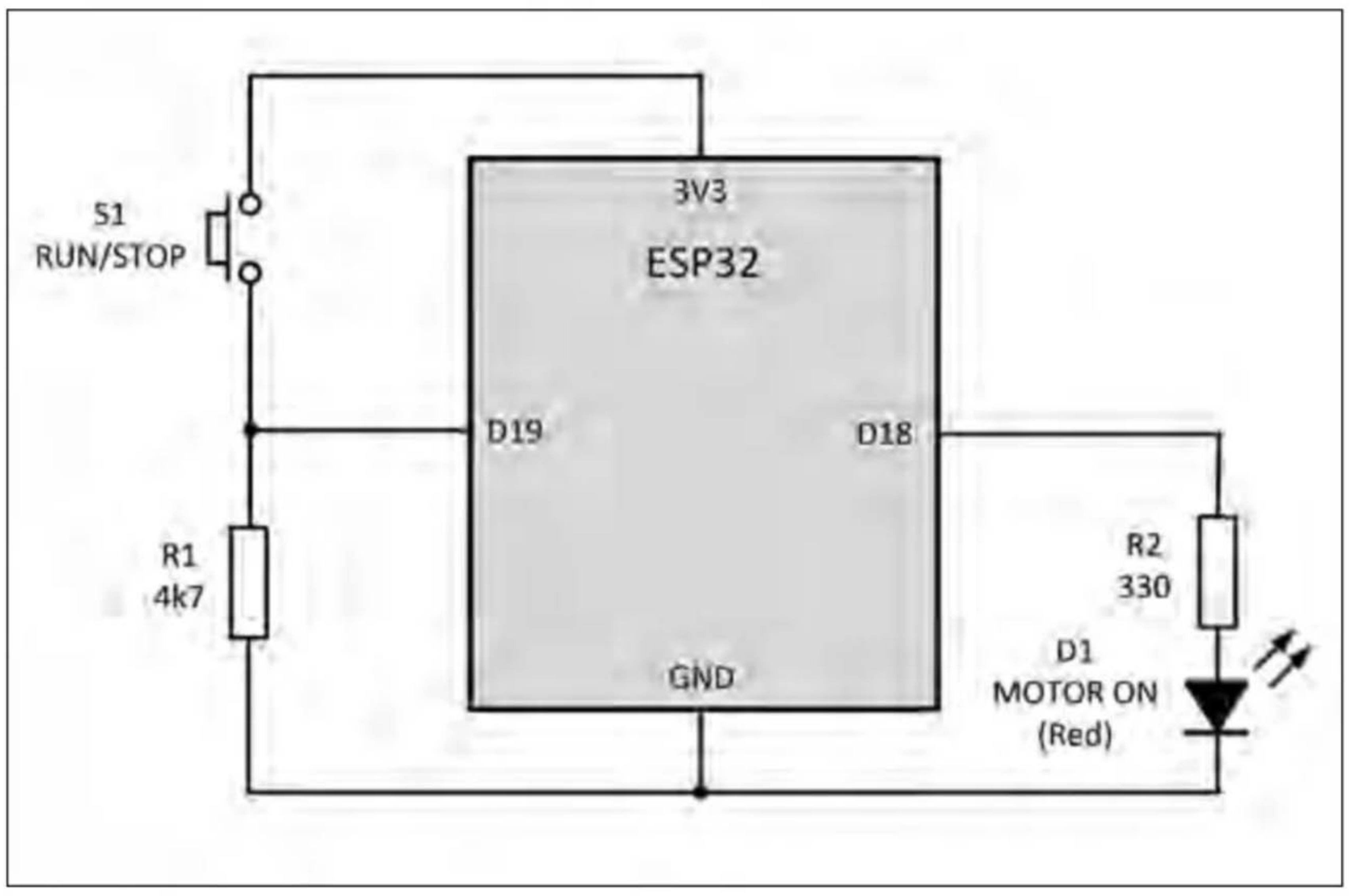


Fig.2.25. Motor controller with toggle action RUN/STOP button.

Next, follows code that’s used for setting up. This code is placed in a function called `setup()` and it is executed at the beginning and only once. The `setup()` function is often used to specify the pin modes (ie, input or output) and to configure the IDE’s Serial Monitor, but if they are not being used then the `setup()` function can simply be left empty.

The main program code is written inside a loop that executes forever (or until the reset button is pressed or the power is removed). This `loop()` function contains the functions that will execute when the program is being run. Each function takes the form of a block of code that is executed whenever the function is *called*. Functions can be the ones that are built into the language, or they can be user-defined. This feature allows us to extend the basic language for our own needs with our user-defined functions calling other functions (both user-defined and in-built) as and when required. Function declarations take the form of one or more statements enclosed between curly braces, { and }. Note that each individual statement *must* end with a semicolon, `;`.

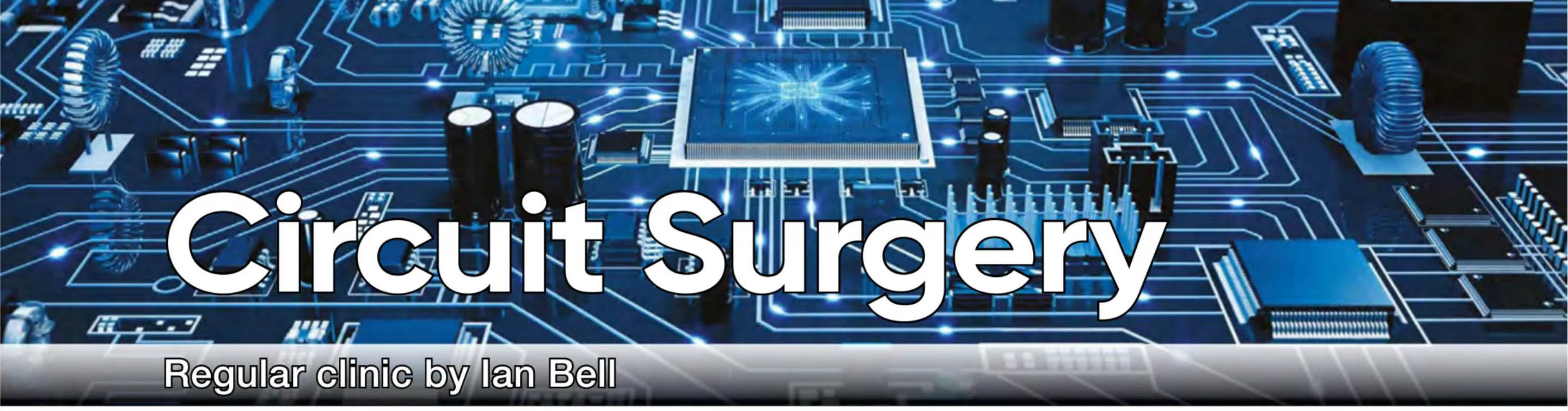
As well as the block comments that we mentioned earlier, comments can be placed *in-line*. These consist of plain text appearing after two `//` characters and added at the end of the line to which they apply. As previously mentioned, comments provide us with a useful reminder of what’s going on in the code and they can be invaluable when maintaining and debugging a program. Indenting (often by three or four spaces) is used to assist with program readability. This becomes particularly important when functions are enclosed within other functions. Top tip – do note that the IDE can tidy up your code very nicely if you select Tools and then Auto Format.

Teach-In challenge

Our second *Teach-In* challenge is one that will test your coding skills. Fig.2.25 shows a simplified version of the motor controller in which only one button is provided. This button is to have a toggle function; press once for RUN and then again for STOP. See if you can modify the code from Listing 2.4 to achieve this.

Next month

Next month, we will move into the analogue world and investigate the ESP32’s analogue input and output capability. We will be introducing techniques for generating analogue waveforms, and our *Practical Project* will feature a simple battery checker. Coding workshop will provide an explanation of the binary, octal and hexadecimal number systems.



Circuit Surgery

Regular clinic by Ian Bell

LTspice 24 and Frequency Response Analysis – Part 2

Last month, we started looking at the frequency response analysis (FRA) function in LTspice, which was part of a major update from LTspice XVII (17.0) to LTspice 17.1. Since then, there has been another major update with a new version number: LTspice 24. Although our main focus in these articles is the FRA, like last month, we will start by looking at some of the changes in this release.

User interface changes

Unlike the LTspice 17.1 update, in the move to LTspice 24 there is a very obvious visual difference, as icons and cursors have been updated (see Fig.1 and Fig.2 which show the old and new toolbars respectively). The default background has also changed.

The new toolbar contains more or less the same functions, but the order has changed to some extent, and there are some items which have been added or removed. For example, the 'Close All' windows button has gone, but there are now both 'Tile Horizontally' and 'Tile Vertically' buttons, previously there was just a button to tile windows horizontally. There is a new 'Configure Analysis' button (to set up the simulation). Some functions have been renamed: 'Control Panel' is now the more conventional 'Settings', and schematic editing 'Drag' mode is now called 'Stretch'. This is where components are moved without breaking the wiring connections, which is useful for adjusting the layout of a schematic. The general aim of these changes seems to be to update to more modern conventions in function naming and icon style.

Some default keyboard short cuts have changed; for example, schematic editing duplicate mode was F6 and is now Ctrl+C. There is a new 'Keyboards Shortcut Cheat Sheet' window which can be opened as a reference for the shortcuts. This is a non-modal window, so you can keep it open for reference as you work. It is possible to revert to the old ('LTspice Classic') shortcuts via the edit shortcuts function that is available from this window – useful if the old ones are too ingrained and you don't want to relearn!

In the old LTspice, the simulation configuration window (which was called 'Edit Simulation Command', and is now called 'Configure Analysis') would forget details of commented-out simulation directives. Now, if there is appropriate text on the schematic, the relevant parameters will be shown in the appropriate tab in the simulation configuration window when it is opened. Also, using shift-click, it is now possible to toggle text on the schematic between being a SPICE directive (such as a simulation command) and a comment (which LTspice will not act on as a command). This is very useful if you have a few simulation commands set (eg, for transient and AC analysis) as it provides a quick way to switch between them. You can tell these apart in the default colour scheme – directives are black, and comments are blue.

The are some useful changes to the waveform viewer. Plot panes can now be moved around and reordered more easily using move up/down functions from the right click menu. All cursors can be removed using the Esc key, and gain and phase margin annotation can be added where applicable. The help system has also changed and now loads pages via a browser. The Help menu has been expanded and includes Open Examples, which provides quick access to the examples included in the download.

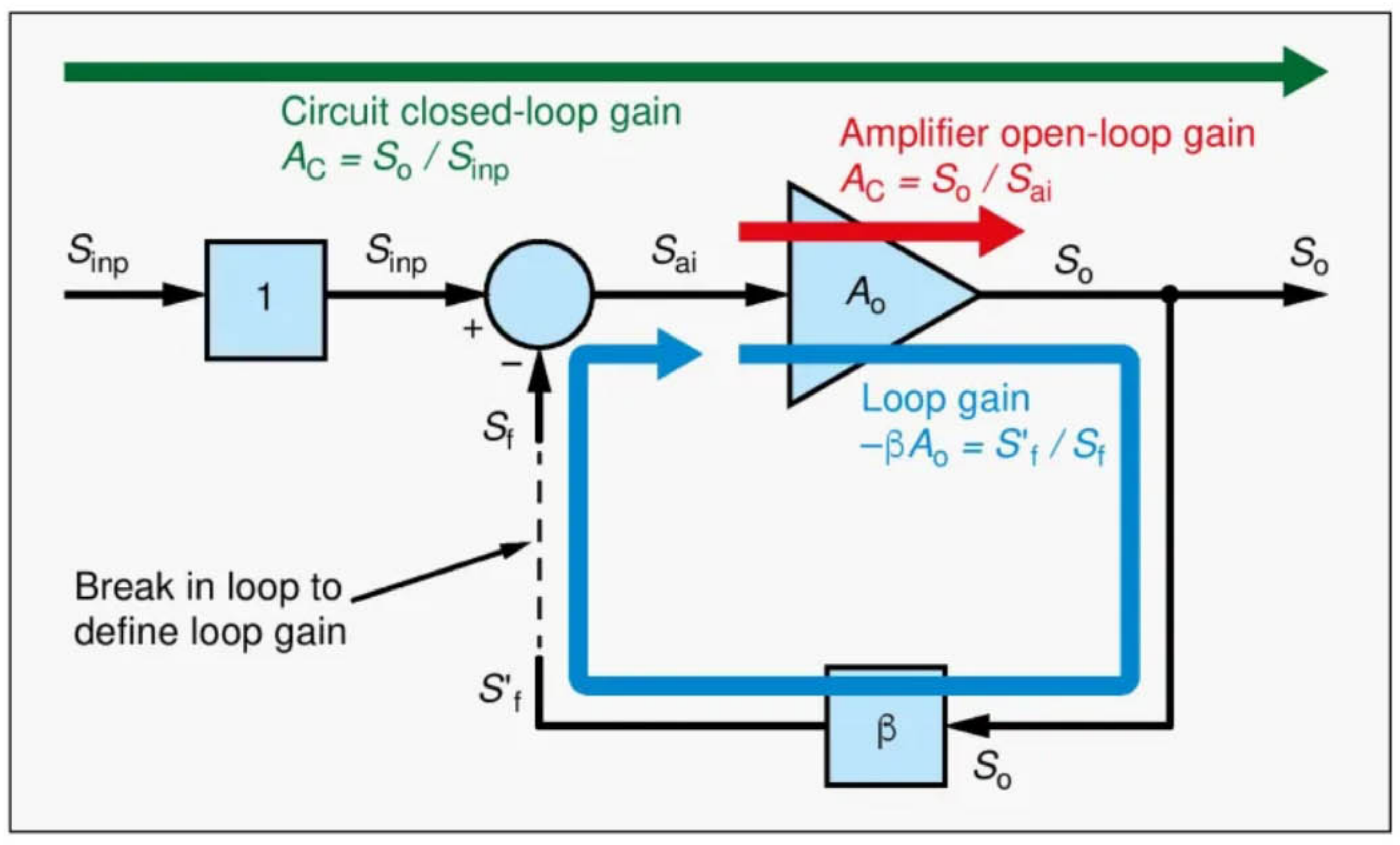


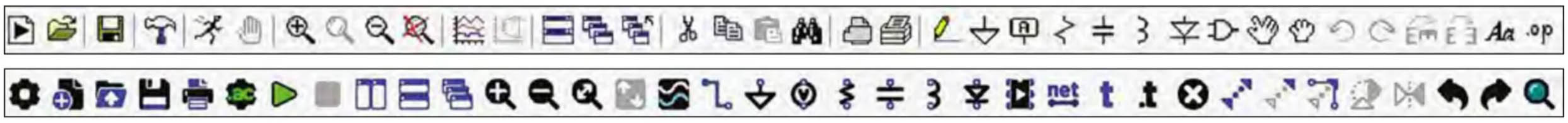
Fig.3. Structure of an amplifier with negative feedback showing the open-loop (red), closed-loop (green) and loop-gain (blue) relationships.

Frequency response analysis recap

LTspice's frequency response analysis (FRA) is aimed at determining the stability of negative feedback circuits; that is, finding out how much margin of error there is between stable and unstable operation. The LTspice FRA is optimised for use with switch-mode powers supplies (SMPS) but it can be used with other feedback circuits. To introduce the basic concept and use of the FRA, last month we started looking at applying the FRA to simple op amp circuits rather than dealing with the complexities of an SMPS.

Last month, we covered some basics of feedback analysis of op amp amplifiers (see Fig.3, which shows a system diagram of a noninverting op amp amplifier). Key concepts are open-loop gain (the gain of the op amp with no feedback applied, A_O) the fraction of signal fed back (β), closed-loop gain (the gain of the circuit with feedback, A_C) and loop gain. Loop gain is the gain around the feedback loop ($-\beta A_O$) and is an important parameter when considering the stability of feedback systems.

Measuring loop gain for a real circuit, and in simulation, is tricky as it



(Above) Fig.1. Old LTspice Toolbar; (Below) Fig.2. Updated toolbar in LTspice 24.

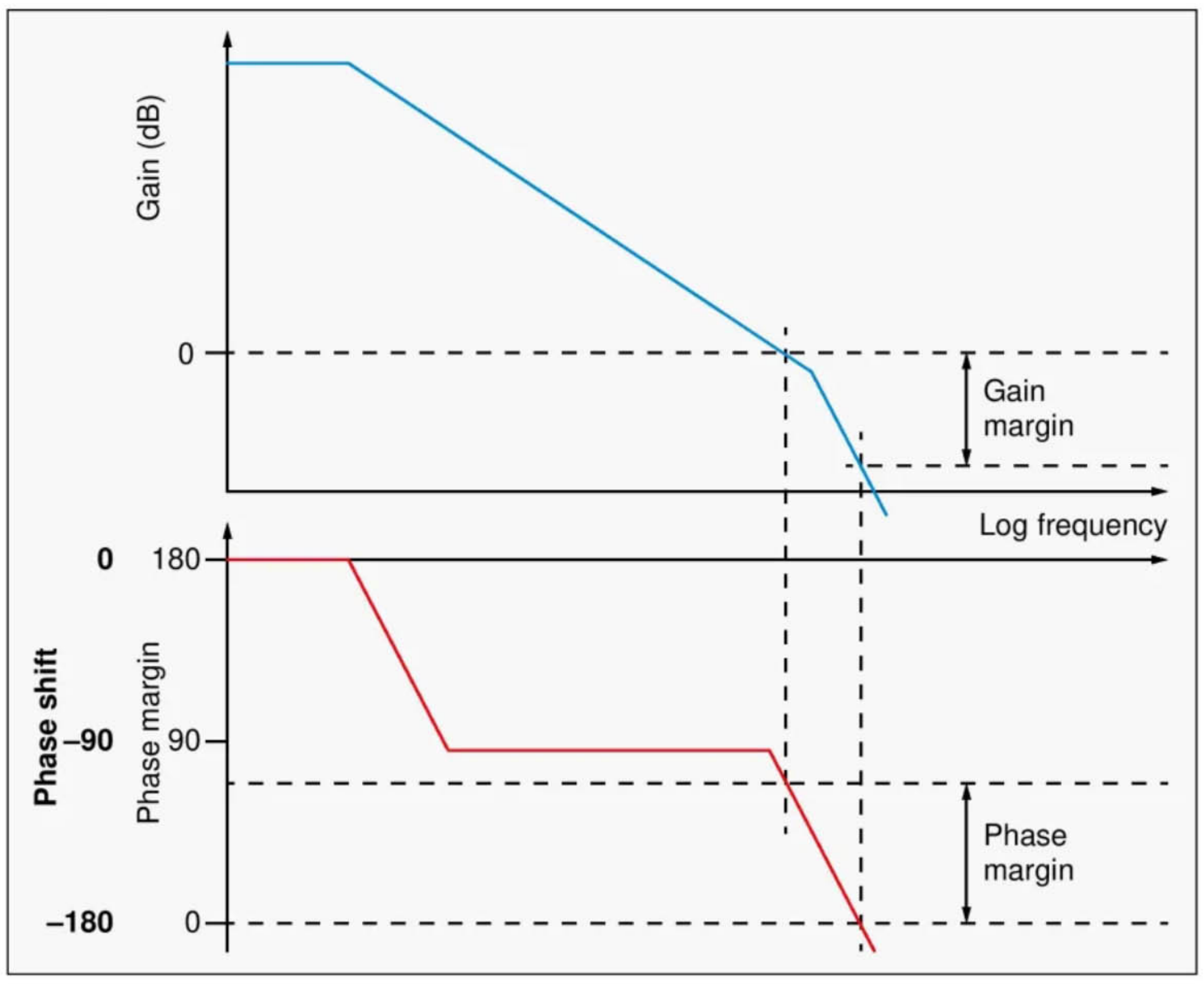
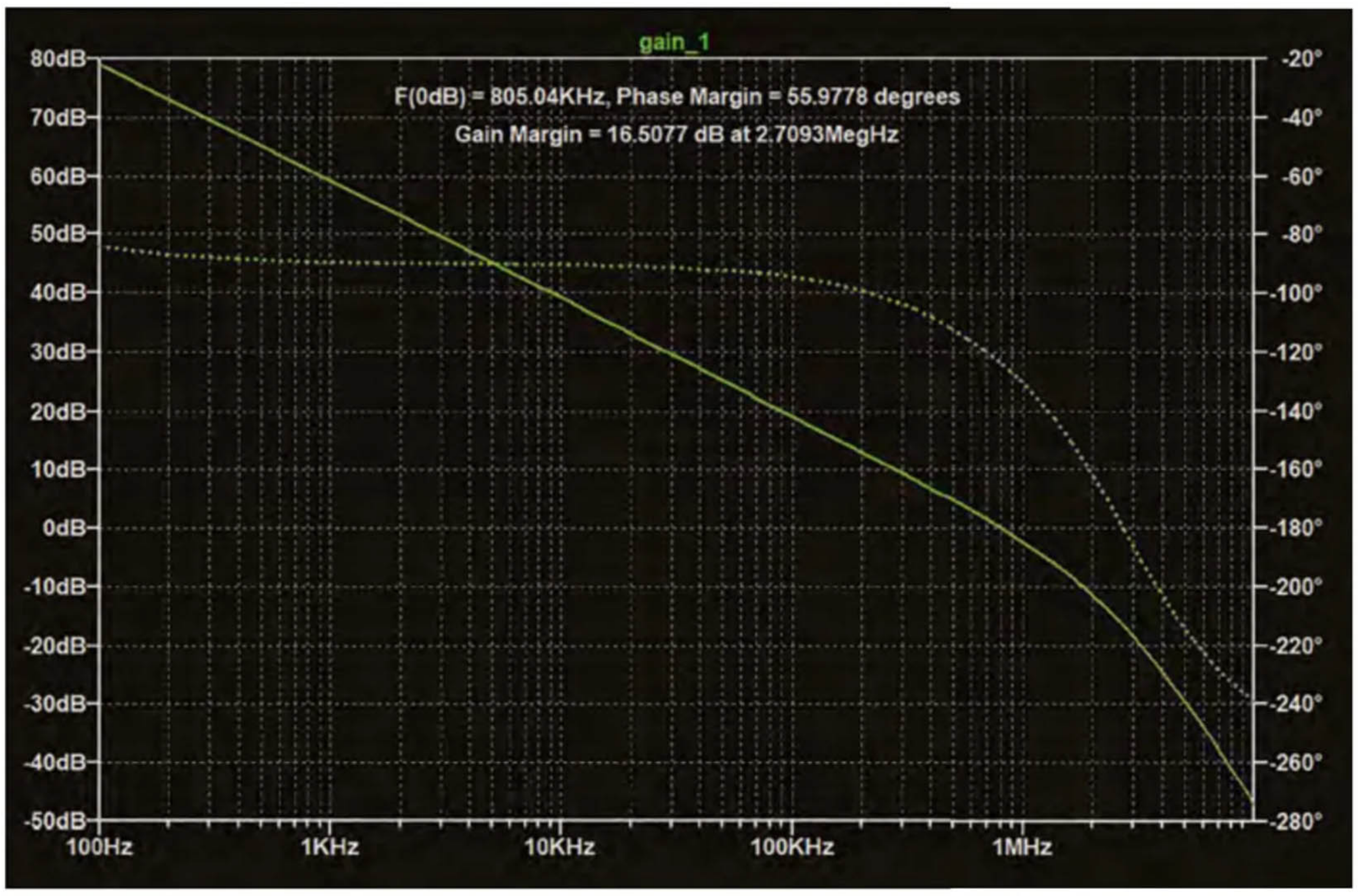


Fig.4. Loop gain plot showing gain margin and phase margin. The phase axis can be labelled with either phase shift or phase margin.

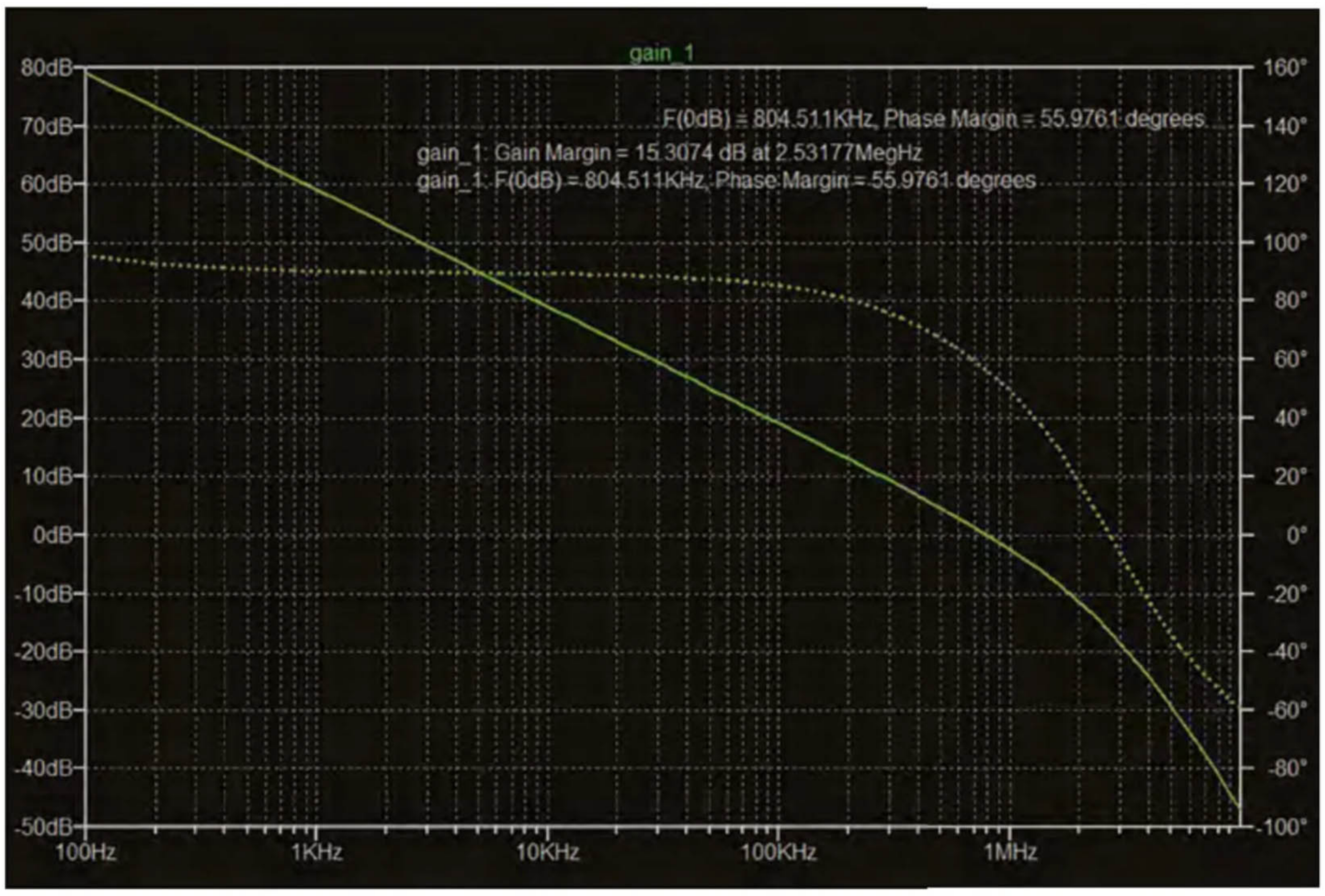
potentially involves 'breaking the loop', which may profoundly affect a circuit's function. Fortunately, a number of techniques have been developed to help overcome this problem, including the Middlebrook method (published in the paper 'Measurement of Loop Gain in Feedback Systems', in the *International Journal of Electronics* in 1975), which is implemented by LTspice in its FRA. The result of the LTspice FRA is a plot of loop gain and phase against frequency and reported values for gain margin and phase margin. These indicate how much additional gain at 180° phase shift, and how much additional phase shift at gain 1 (0dB) would be needed to cause instability (see last month for details). These values indicate how stable the circuit is likely to be in operation.



F(0dB) = 805.04KHz, Phase Margin = 55.9778 degrees

Gain Margin = 16.5077 dB at 2.7093MegHz

Fig.5. Example FRA results from last month using LTspice 17.1.



F(0dB) = 804.511KHz, Phase Margin = 55.9761 degrees

gain_1: Gain Margin = 15.3074 dB at 2.53177MegHz

gain_1: F(0dB) = 804.511KHz, Phase Margin = 55.9761 degrees

Fig.6. FRA results from the example from last month using LTspice 24 with phase and gain margin annotations added. Note the phase margin scale on the phase axis.

FRA updates in LTspice 24

Gain and phase margin are shown graphically in Fig.4. The loop phase can be plotted simply as phase shift (as it would be in an AC analysis) or it can be shown as phase margin by setting the axis scale values to phase shift + 180°, which makes it easier to read the phase margin (at the 0dB frequency). This change in axis labelling has been made in the FRA in LTspice 24, which now plots phase margin rather than phase shift. This can be seen in Fig.5 and Fig.6 which show the example FRA plot from last month (with LTspice 17.1) and the results from the same simulation run in LTspice 24.

Also, when this example was run in LTspice 24 it did not automatically add text giving both the phase and gain margins (as seen in Fig.5), it just showed F(0dB) and the phase margin, however, gain and phase margin annotations can be added manually by using the right-click menu on the waveform plot and selecting 'Notes & Annotations' and then 'Annotate Gain Margin' (or phase margin). These added annotations are included in Fig.6.

As explained last month, running an FRA requires an FRA component to be added to the schematic, specifically inserting it into the feedback loop. Other enhancements to the FRA in version 24 include a new 4-terminal probe (schematic component) which can be used in some situations where the FRA analysis was difficult to implement with just the FRA component. Analogue Device's release notes also say that the stimulus generated to run the FRA has been improved to provide better accuracy – it now has smoother changes between waveforms as frequency is changed.

Transient analysis frequency response

LTspice users and regular readers of *Circuit Surgery* will be familiar with plots of gain and phase against frequency obtained using AC analysis. FRA plots looks similar to AC analysis results; however, the simulation is performed in a fundamentally different way. The setup required for a transient-based analysis is more complex than for AC analysis, so we'll discuss the issues involved before describing the FRA settings in LTspice.

AC analysis uses a linearised model of the circuit, from which LTspice can rapidly calculate the response at any frequency. Strictly speaking, AC analysis only works for very small input amplitudes (AC analysis is also called

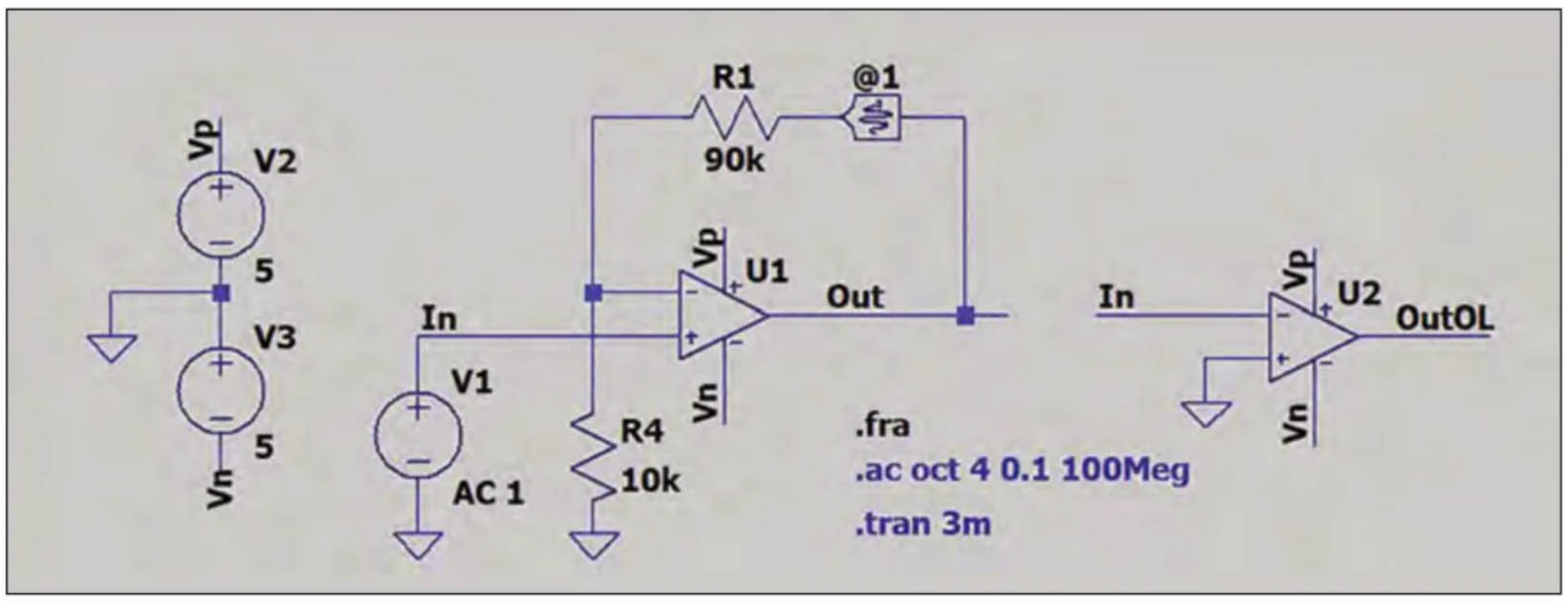


Fig.7. LTspice for investigating FRA with a simple op amp amplifier.

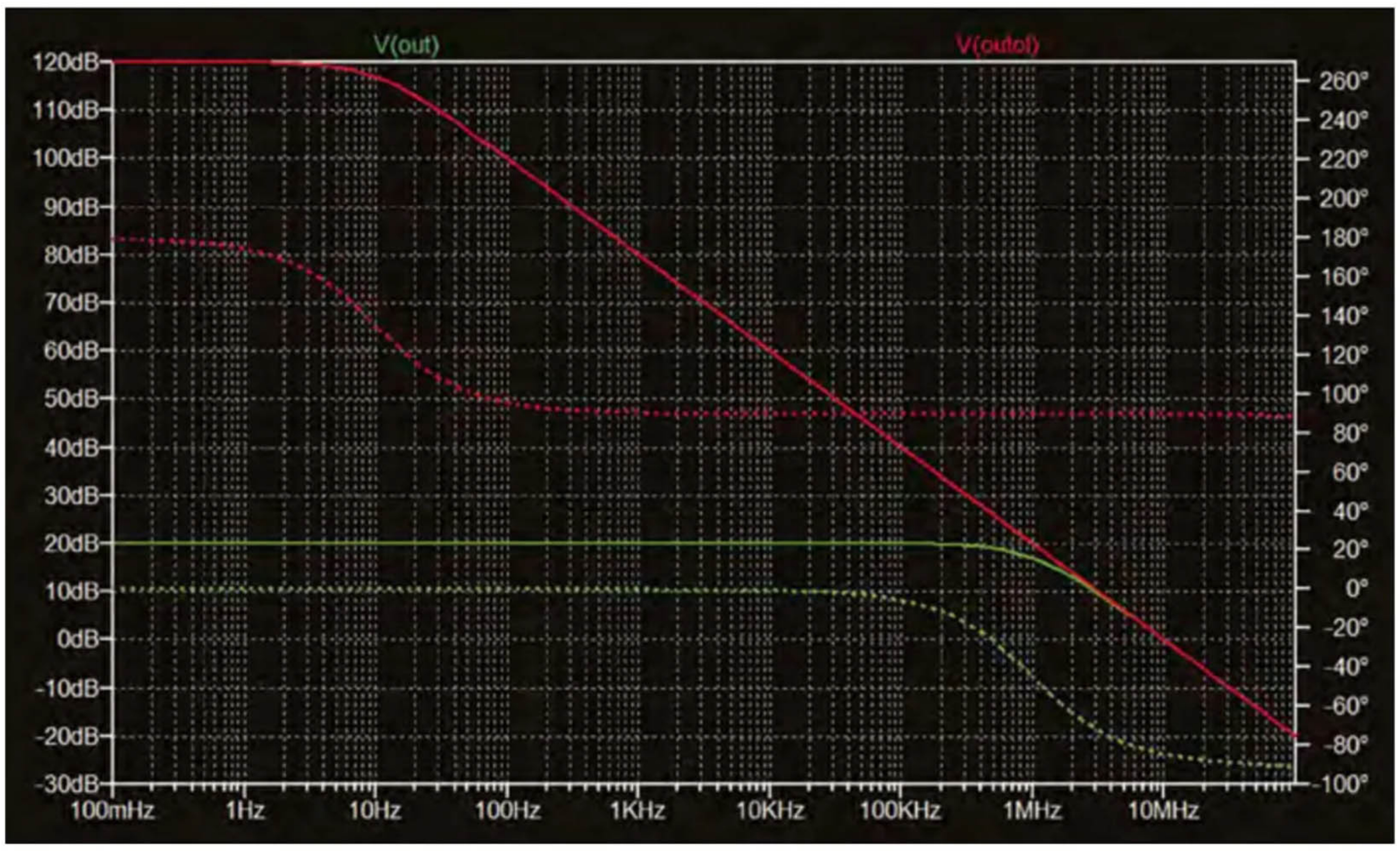


Fig.8. AC analysis results (closed-loop circuit and op amp open loop) from the Fig.7 circuit.

'small-signal analysis'). The model used only applies to a particular operating point (the DC conditions, or bias, in the circuit). Large signals will shift things too far from this point, invalidating the model parameters. For example, the gain of an amplifier will decrease at high input levels as it starts to clip. However, once created, the linearised model gives the same gain *vs* frequency result for any amplitude input. Therefore, it is common to use a 1V input from AC analysis in LTspice even if the real circuit would not work correctly with this signal level. With an input of 1 the output is numerically equal to gain and therefore the gain is easy to plot – you just plot the signal value (voltage); you don't have to calculate the gain as `v(out)/v(in)`.

LTspice's FRA is aimed at stability analysis of SMPSs, which are non-linear circuits with no small-signal linear equivalent circuit that could be used for AC analysis. However, there is an alternative to AC analysis, which is to directly measure the gain and phase shift with a range of sinewave stimuli, using transient simulation (time-based simulation to obtain circuit waveforms). This is the approach used in LTspice's FRA.

Transient simulation corresponds with what would be done with a real circuit in the lab to obtain a frequency response – typically using a signal generator and oscilloscope, often using automated measurements to cover the many different frequencies required to calculate gain and phase shift. This is straightforward for a simple input to output response, but as discussed last month, measuring loop gain requires special techniques such as those published by Middlebrook. The Middlebrook approach was developed for use in the lab but can also be implemented using transient simulation. The LTspice FRA runs a transient simulation in which it injects signals into the feedback loop and analyses the response in accordance with a version of Middlebrook's method.

Setting up an LTspice FRA

Setting up an AC analysis for input to output gain is straightforward – it just requires the input amplitude (usually 1, as noted above) from a standard source at the circuit input, and the required frequency range and number of points for the plot to be produced. For a transient simulation-based frequency response there is more to think about. Signal amplitudes must be suitable – too high and the circuit will be forced outside its normal operating conditions, for example clipping, or slew rate limiting in the case of an amplifier – too low and the output / response signal may be difficult to measure / resolve due to noise (in the lab) or accuracy limitations (in simulation).

Because gain changes with frequency (the whole point of performing a frequency response analysis) the suitable signal levels may also change with frequency. Also, when the frequency is changed during the analysis run any resulting abrupt changes in the input / injected waveform may cause responses such as spikes that could temporarily invalidate the analysis. Therefore, it may be necessary to wait for the circuit to stabilise after each frequency step. Furthermore, the circuit may take time to settle to normal operation at start up, so an initial delay before analysis starts may be required.

For the specific case of loop-gain analysis, it is also necessary to decide where in the loop the signal will be injected. In the case of LTspice this involves adding an FRA component to the schematic (this has name prefix '`@`'). The analyser component applies the sinusoidal stimuli and measures the response. To reduce simulation time, it may optionally simultaneously inject multiple harmonics, although this may reduce accuracy.

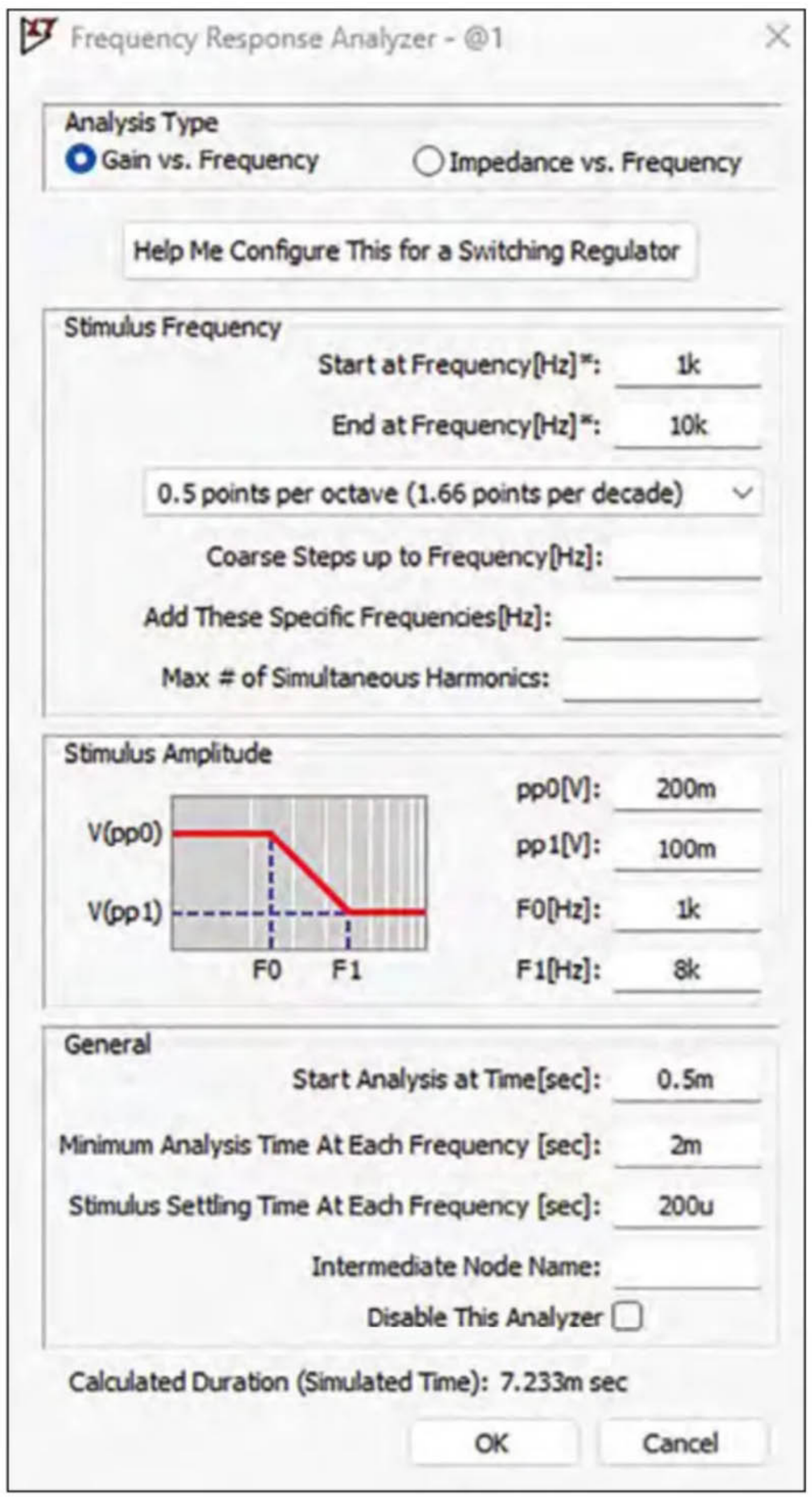
The LTspice FRA uses the voltage-only version of Middlebrook's method, which requires the device to be inserted into a point where a low impedance is driving a high impedance. For an op amp amplifier, the op amp output and feedback resistor should be suitable. For an SMPS, the FRA component would usually be placed between the power supply output and error amplifier input. The FRA must interrupt all paths in the loop – that is, there must not be a parallel path in the loop which bypasses the FRA component.

Example circuit

The circuit in Fig.7 will be used to look at basic use of the LTspice FRA and relate the results to the op amp theory we discussed last month. The circuit

Introduction to LTspice

Want to learn the basics of LTspice? Ian Bell wrote an excellent series of *Circuit Surgery* articles to get you up and running, see *PE* October 2018 to January 2019, and July/August 2020. All issues are available in print and PDF from the *PE* website: **https://bit.ly/pe-backissues**


Frequency Response Analyzer - @1

Analysis Type

● Gain vs. Frequency ○ Impedance vs. Frequency

Stimulus Frequency

Start at Frequency[Hz]*: 1k

End at Frequency[Hz]*: 10k

0.5 points per octave (1.66 points per decade)

Coarse Steps up to Frequency[Hz]:

Add These Specific Frequencies[Hz]:

Max # of Simultaneous Harmonics:

Stimulus Amplitude

pp0[V]: 200m

pp1[V]: 100m

F0[Hz]: 1k

F1[Hz]: 8k

General

Start Analysis at Time[sec]: 0.5m

Minimum Analysis Time At Each Frequency [sec]: 2m

Stimulus Settling Time At Each Frequency [sec]: 200u

Intermediate Node Name:

Disable This Analyzer ☐

Calculated Duration (Simulated Time): 7.233m sec

Fig.9. FRA setup for trial run.

uses an idealised op amp (LTspice `UniversalOpAmp2`, and is a basic non-inverting amplifier with a gain of 10 (20dB) ($1+R_1/R_2$ = 1 + (90kΩ /10kΩ) = 10). The FRA component (`@1`) is inserted between the op amp `U1` output (low-impedance op amp output) and feedback resistor (relatively high impedance since 90kΩ is used) to fit the requirements for FRA insertion. The schematic has three analyses configured on it (FRA, AC and transient). The LTspice 24 update means it is now easy to switch between these using Shift-Click, which toggles between SPICE directive (black text) and comment (blue text).

Op amp `U2` is the same as `U1` and is included to provide an open-loop AC analysis. This is done in a crude way here – simply applying the input to the open-loop amplifier. This works for this idealised device in simulation, but for real devices (particularly in the lab) more sophisticated approaches would be needed; for example, to ensure that the device is DC stable and does not saturate.

Fig.8 shows the results of running the AC analysis with the FRA device deactivated. It is helpful to understand this before looking at the FRA results. The idealised model for `U1` is configured to have an open-loop gain of one million (120dB). This value can be changed by right-clicking the op amp symbol (`Avol` parameter). We see the 120dB gain on the open-loop plot (red trace) at low frequencies, but the gain starts reducing above about 1Hz. Specifically, it is down by 3dB (half power) at 10Hz. This is typical of real op amps.

Op amp open-loop gain reduction with increasing frequency is called compensation, and is deliberately designed into the device to help ensure stability when feedback is applied. The frequency at which the gain reduces by 3dB is set by the gain-bandwidth product parameter (`GBW`) in the op amp model, which in this case is 10^7 (10 million). On the sloped part of the open-loop response the product of gain and frequency is 10^7 at all points (eg, 60dB (gain 1000) x 10kHz = 10^7 and 0dB (gain 1) x 10MHz = 10^7).

The closed-loop gain (A_C) is 10 (20dB), as noted above, and we see this gain is maintained for the amplifier circuit until around 1MHz (where it is 3dB down) – we can predict this from the gain-bandwidth product by dividing by the closed-loop gain. The closed-loop bandwidth is GBW/A_C = 1.0 x 10^7/10 = 1MHz.

Last month, we saw that the closed-loop gain equation for a non-inverting op amp amplifier could be simplified by approximating the term (1 + βA_O) to βA_O. This gives the closed-loop gain just in terms of β, which leads to the well-known gain formulae for op amp amplifiers that only depend on the feedback resistors, not on the open-loop gain. A naive interpretation of the gain formula $1+R_1/R_2$ would imply the same gain at all frequencies. However, this is not the case because it depends on the assumption of high open-loop gain (or more accurately loop gain). The approximation breaks down at high frequencies as the open-loop gain (βA_O) reduces and is no longer much greater than one.

Preparing for FRA

FRA uses transient simulation, so it is important make sure the transient simulation is running correctly before starting FRA. For our example op amp circuit, which is close to ideal, this is straightforward, but may be less so for more complex circuits such an SMPS. This simulation should be used to determine the time the circuit takes to settle to normal operation (and other parameters for an SMPS). If the delay is significant, it should be reduced if possible, or steps taken to speed up the simulation. For the op amp circuit, correct operation can be confirmed by using a sinewave input from source `V1`, but if this is done the source function should be changed from `sine` to `none` before the FRA is run to prevent the signal from interfering with the FRA later. If not already in place, the FRA component should be added to the schematic at a suitable point in the feedback loop. If it is in place already then it needs to be disabled during the standard transient simulation.

Full FRA simulations can be quite long, so it is a good idea to start with a trial run that uses a very small number of frequencies to check it is operating correctly. The main setup for the FRA analysis is done via the FRA device right-click menu, rather than through the more usual use of the Analysis Configuration dialogue.

FRA device configuration and trial run

Fig.9 shows the setup dialog for the FRA device configured for a trial run for the circuit in Fig.7. There are a lot of parameters which can be set for the FRA. We will discuss the parameters used for the initial run now, and comment on others later.

The FRA can perform gain or impedance *vs* frequency analysis – we are looking at gain, so this is selected. The start and end frequencies are obvious parameters, but their required values depend on the properties and specification of the circuit being analysed. For a trial run these should be fairly close together to keep the number of frequencies analysed small. The number of frequencies is also controlled by the points per octave drop-down. A low setting (0.5) should be used for trial runs. More points give a more detailed graph but will take longer.

As noted above, the stimulus amplitude must be suitable for the circuit and may need to vary with frequency. The FRA setup allows you to define two frequency corners (`F0` and `F1`) at which the peak-to-peak signal voltages are defined: pp0 (low-frequency amplitude) and `pp1` (high-frequency amplitude) respectively. The amplitude is `pp0` below `F0` and `pp1` above `F1` and varies from `pp0` to `pp1` between `F0` and `F1`. This is the amplitude at the point where the FRA is inserted, which in this case is the op amp output. As the gain of the amplifier is higher at lower frequencies we would typically expect higher amplitudes at low frequencies, but the specific values used here were mainly set for purposes of illustration.

The General section of the dialog contains timing parameters. The start analysis time controls when the first stimulus is applied and should provide enough time for the circuit to settle after the start of the simulation. This is important for circuits such as an SMPS, but the value used here is for illustration rather than being necessary for analysis of the op amp circuit. The minimum analysis time sets the minimum time for applying each frequency. Analysis starts after the settling time – which can

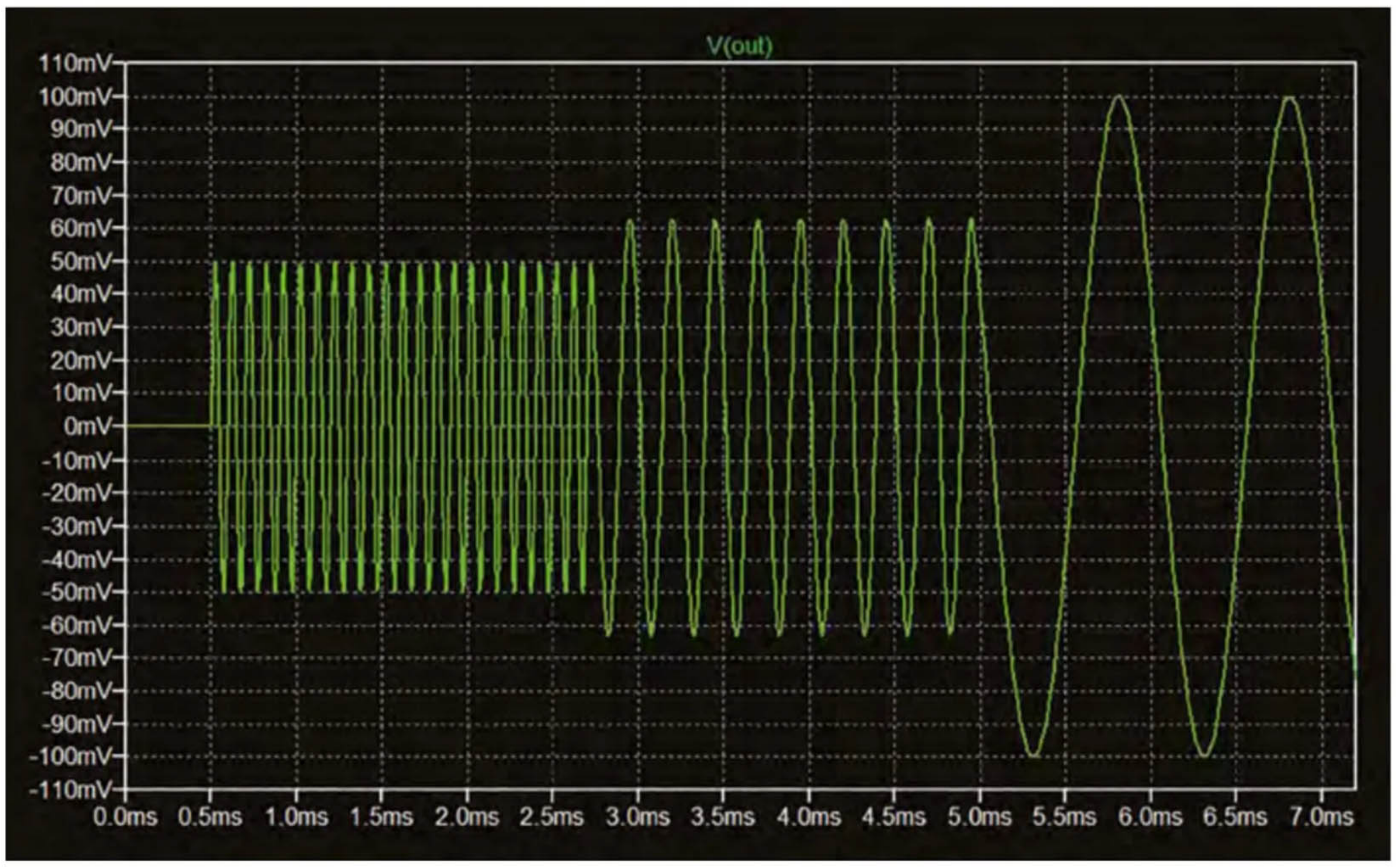


Fig.10. Stimulus waveform for FRA trial run using settings shown in Fig.9.

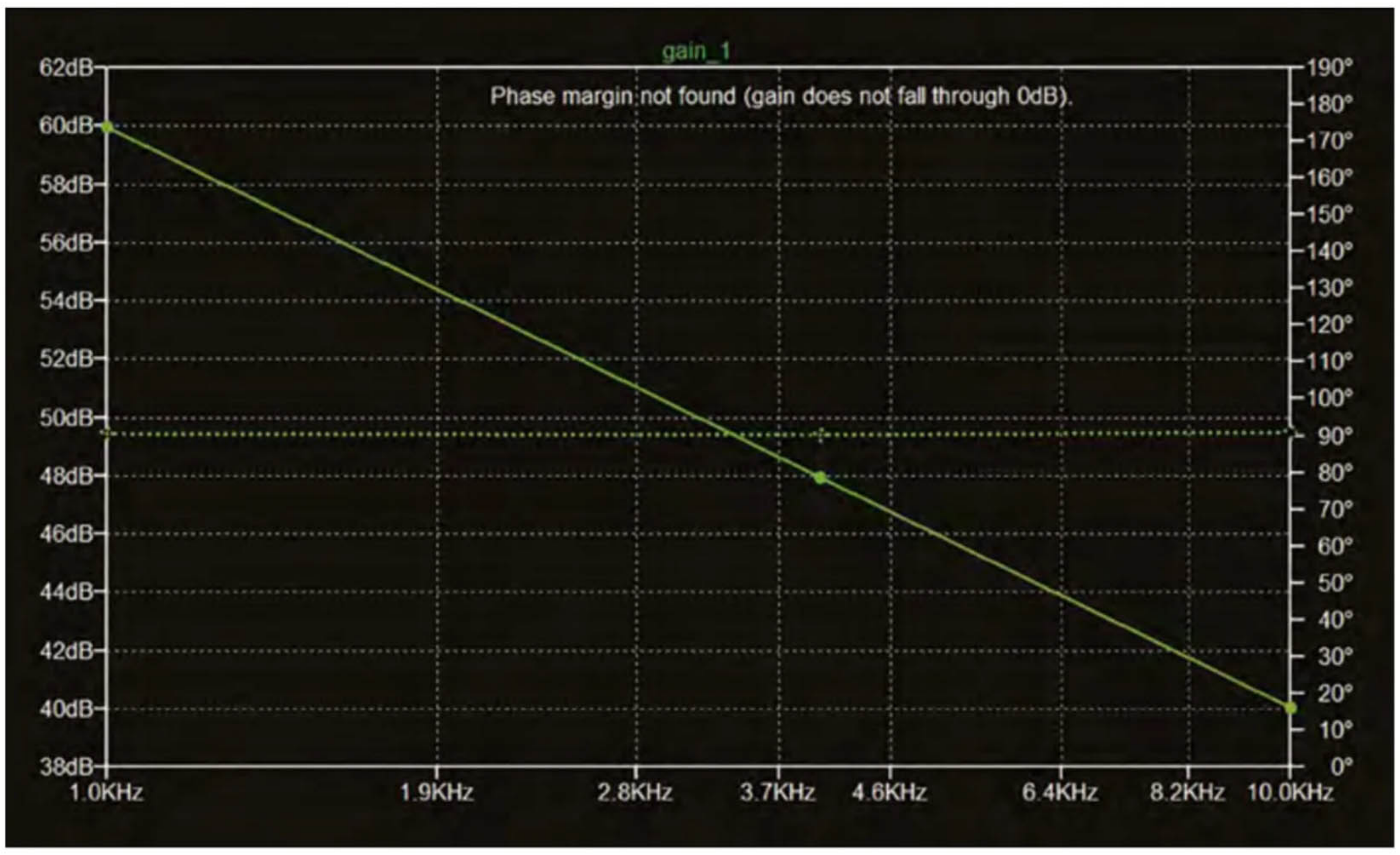
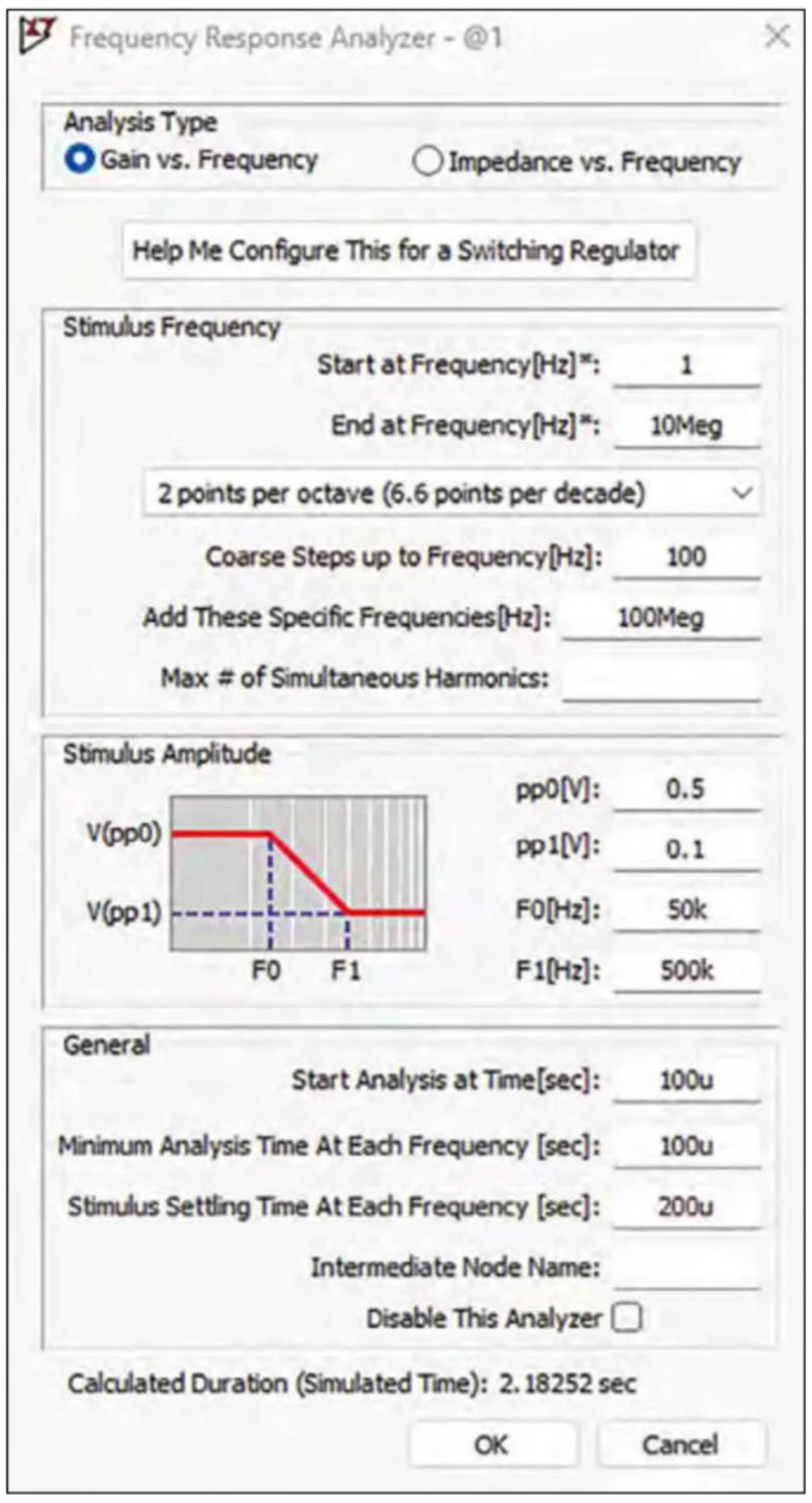


Fig.11. FRA plot for the circuit in Fig.7 using settings shown in Fig.9.


Frequency Response Analyzer - @1

Analysis Type

● Gain vs. Frequency ○ Impedance vs. Frequency

Help Me Configure This for a Switching Regulator

Stimulus Frequency

Start at Frequency[Hz]*: 1

End at Frequency[Hz]*: 10Meg

2 points per octave (6.6 points per decade)

Coarse Steps up to Frequency[Hz]: 100

Add These Specific Frequencies[Hz]: 100Meg

Max # of Simultaneous Harmonics:

Stimulus Amplitude

pp0[V]: 0.5

pp1[V]: 0.1

F0[Hz]: 50k

F1[Hz]: 500k

General

Start Analysis at Time[sec]: 100u

Minimum Analysis Time At Each Frequency [sec]: 100u

Stimulus Settling Time At Each Frequency [sec]: 200u

Intermediate Node Name:

Disable This Analyzer ☐

Calculated Duration (Simulated Time): 2.18252 sec

Fig.12. FRA setup for full run.

be adjusted appropriately if behaviours such as spikes and ripples are observed in the waveforms when the stimulus frequency changes. These parameters are not particularly critical for the op amp example, but for circuits such as an SMPS the values should ensure that sufficient switching cycles are included without being so long as to cause excessive run time. The General section also contains a checkbox for disabling the FRA device – this should be used when other types of analysis are run.

The stimulus waveform for running the FRA for the circuit in Fig.7 with settings in Fig.9 is shown in Fig.10. The frequency response plot is shown in Fig.11. FRA waveforms are plotted in the same way as a normal transient simulation and the response plot window appears automatically after the simulation completes. Three frequencies were used: 1kHz, 4kHz and 10kHz. Ideally, the response plot would include the 0dB gain frequency, but here the values are just to produce illustrative waveforms. The waveform plot shows the 0.5ms delay before the stimulus is applied; the 50mV high-frequency amplitude (at 10kHz); the 100mV low-frequency amplitude (at 1kHz); and an intermediate amplitude at the frequency between these – corresponding with the setting shown in Fig.9. The waveforms look OK (there is no distortion) which indicates that the parameters are suitable for a full analysis.

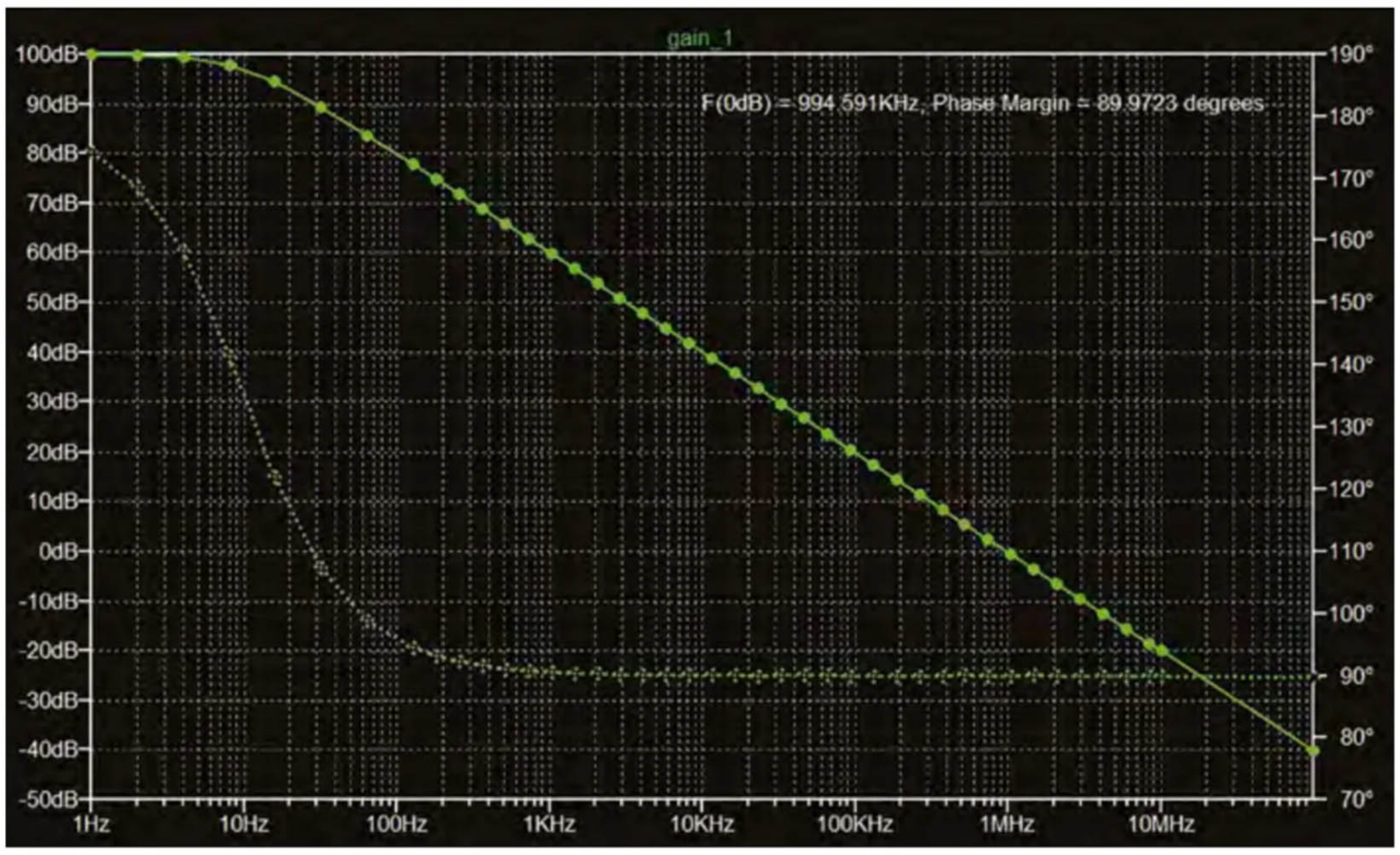


Fig.13. FRA plot for the circuit in Fig.1 using settings shown in Fig.12.

Simulation files

Most, but not every month, LTSpice is used to support descriptions and analysis in *Circuit Surgery*.
The examples and files are available for download from the *PE* website:
https://bit.ly/pe-downloads

Full FRA analysis

The setting used for a full analysis are shown in Fig.12 and the results in Fig.13. The frequency range has been increased to cover a wider range, similar to that in Fig.8, with more points per octave than the trial run. The range goes up to 10MHz with an additional specific frequency of 100MHz, which extends the range plotted without requiring many more waveform steps. The datapoints are marked on Fig.13, which shows the six or seven points per decade stopping at 10MHz, and the single additional point at 100MHz. Furthermore, the coarse stepping parameter is set to 100Hz, which reduces the number of frequency steps per octave below the specified frequency.

These last two parameters are not really needed here but are included to illustrate their use. They can help reduce simulation time. For an SMPS, the simulation time may be long at low frequencies because many switching cycles are required for just one cycle of the stimulus. Simulation time can also be reduced by setting the number of simultaneous harmonics (typically 2 to 4). This applies multiple frequencies at the same time – it is faster, but less accurate.

Comparing the results in Fig.13 with Fig.8 we see that the loop gain (βA_O) is equal to the difference (on the dB scale) between the open- and closed-loop gains, while the loop gain is much larger than 1. This matches theoretical expectations – the relationship can be confirmed mathematically as follows, using the equation for closed-loop gain discussed last month.

$$20\log(A_O) - 20\log(A_C) = 20\log(A_O) - 20\log\left(\frac{A_O}{(1+\beta A_O)}\right)$$

$$= 20\log\left(A_O\frac{(1+\beta A_O)}{A_O}\right) \approx 20\log(\beta A_O)$$

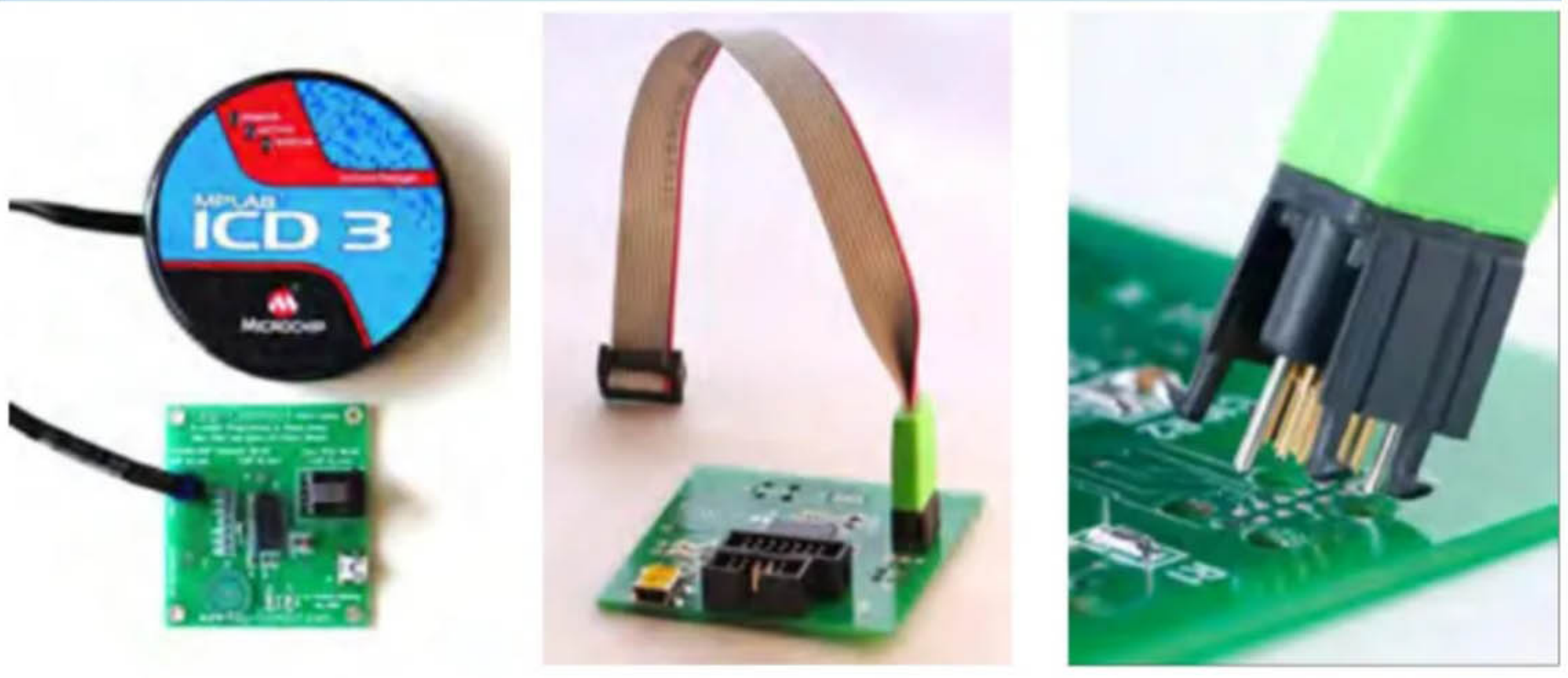
For readers interested in investigating the FRA further, particularly for SMSP circuits, there are additional details available via the LTspice help, and SMPS FRA examples are provided with the LTspice 24 download.

NEW!
5-year collection
2017-2021
All 60 issues from Jan 2017
to Dec 2021 for just £44.95
PDF files ready for
immediate download
See page 6 for further details and
other great back-issue offers.
Purchase and download at:
www.electronpublishing.com

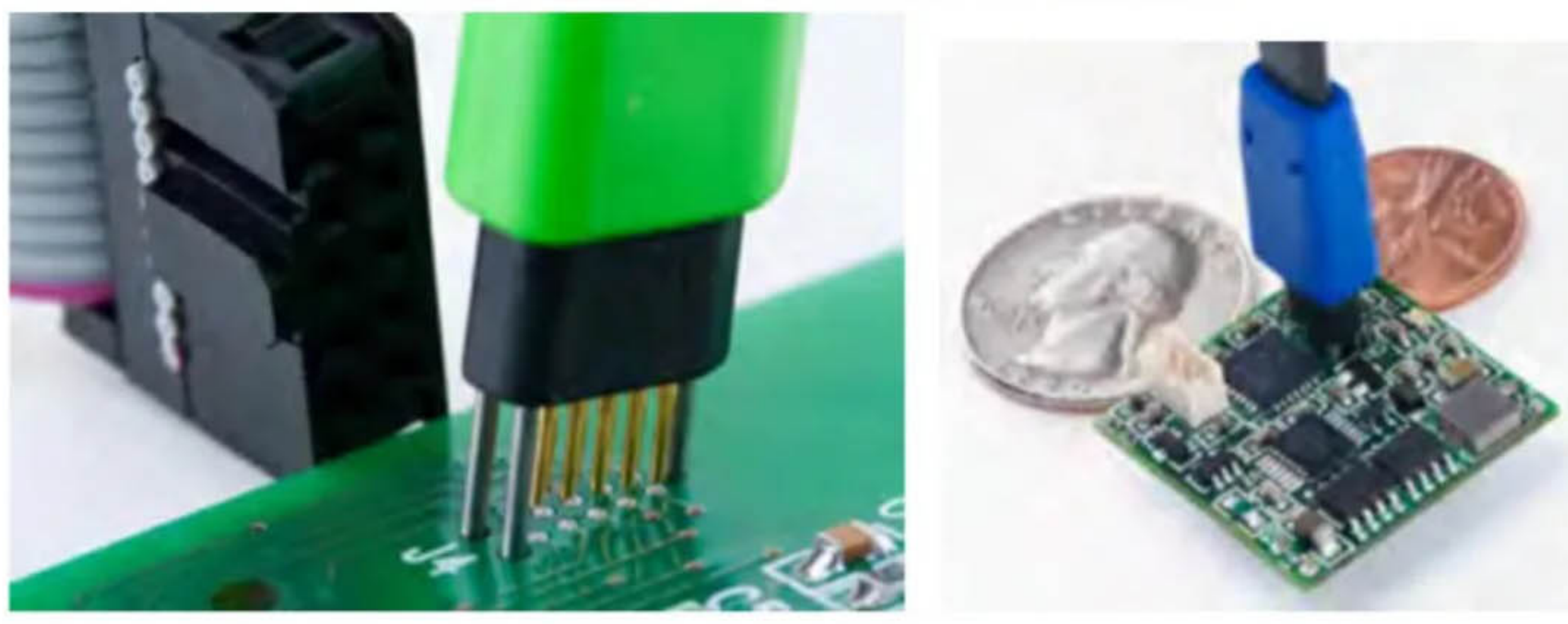
CRICKLEWOOD
ELECTRONICS
Established 1981
Your best bet since MAPLIN
Chock-a-Block with Stock
Visit: www.cricklewoodelectronics.com
Or phone our friendly knowledgeable staff on 020 8452 0161
Components • Audio • Video • Connectors • Cables
Arduino • Test Equipment etc, etc

Visit our Shop, Call or Buy online at:
www.cricklewoodelectronics.com
020 8452 0161
Visit our shop at:
40-42 Cricklewood Broadway
London NW2 3ET

Tag-Connect



JTAG Connector Plugs Directly into PCB!!
No Header! No Brainer!



Our patented range of Plug-of-Nails™ spring-pin cables plug directly into a tiny footprint of pads and locating holes in your PCB, eliminating the need for a mating header. Save Cost & Space on Every PCB!!
Solutions for: PIC . dsPIC . ARM . MSP430 . Atmel . Generic JTAG . Altera Xilinx . BDM . C2000 . SPY-BI-WIRE . SPI / IIC . Altium Mini-HDMI . & More
www.PlugOfNails.com
Tag-Connector footprints as small as 0.02 sq. inch (0.13 sq cm)

Max's Cool Beans

By Max the Magnificent

Arduino Bootcamp – Part 16

Way back in the mists of time we used to call *Part 14* (*PE*, February 2024), we commenced with me complementing you on your perspicacious and sagacious appearance, after which I invited you to say something nice about me.

Well, a reader we will call Ron (because that's his name) rose to the challenge. Ron just emailed me to say: *You look great in that Hawaiian shirt, and I love the way you digress loquaciously with verbosity and circumlocution when talking about things of an electronic nature.* Well done, Ron. Although I don't like to boast (I'm known far and wide for priding myself on my humility), I must admit that I take some satisfaction with respect to the garrulity and circuitousness of my humble scribblings.

Do you know the time?

In our previous column, we added a DS3231 real-time clock (RTC) to our spiffy new dual-breadboard setup. The great thing about an RTC is that its backup battery allows it to remember the time when power is removed – either accidentally or intentionally – from the main system.

Just to remind ourselves of where we're at, you can download an image of our current hardware setup (file **CB-Apr24-01.pdf**). As usual, all the files mentioned in this column are available from the April 2024 page of the *PE* website: **https://bit.ly/pe-downloads**

Next, we created two small programs (or 'sketches' in the vernacular of the Arduino). The first loaded the RTC with the current date and time. The second looped around reading the date and time from the RTC and displaying these values in the Serial Monitor. If you've forgotten what this program looks like, you can download a copy now (file **CB-Apr24-02.txt**).

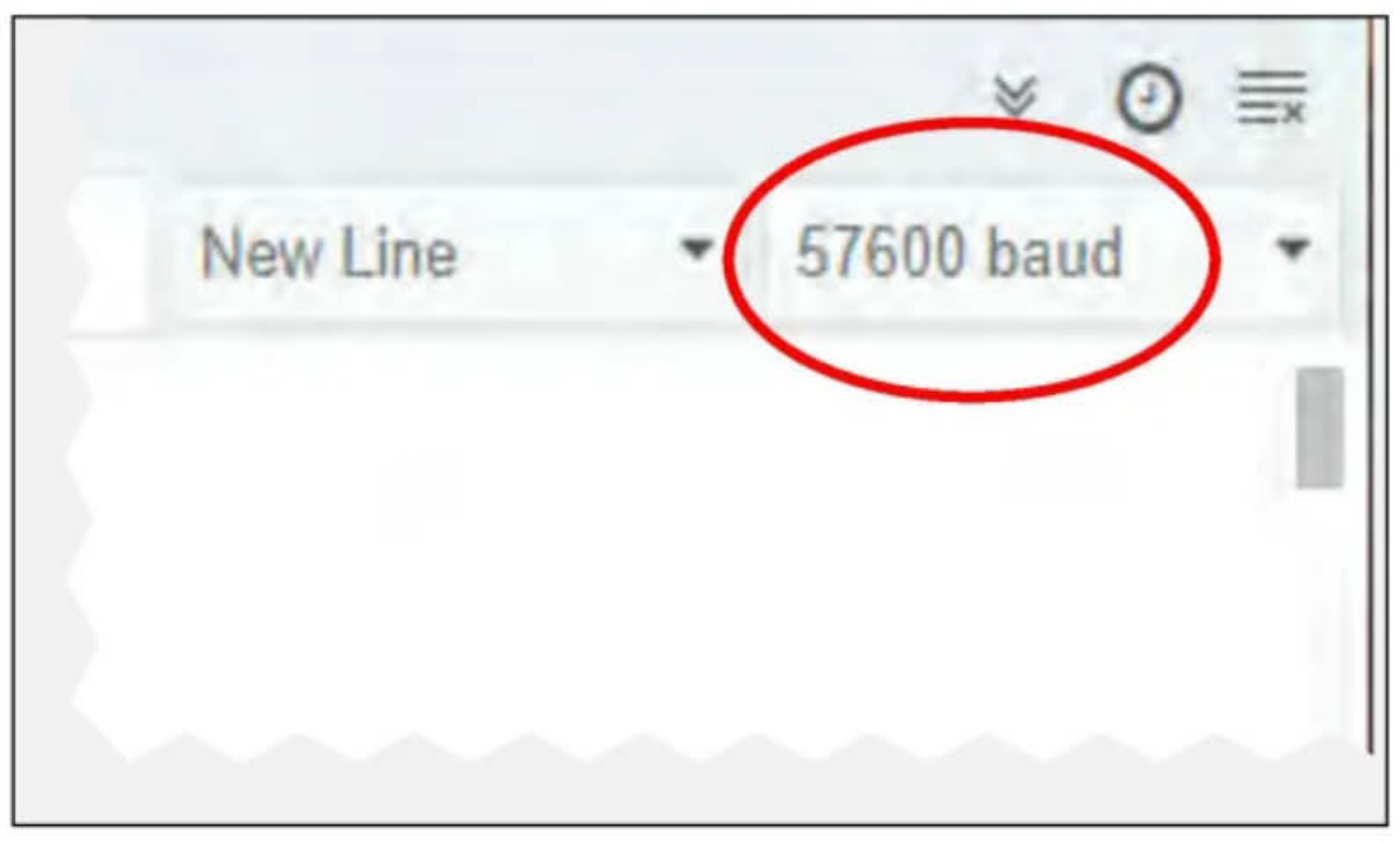


Fig.1. Checking the baud rate in the Serial Monitor.

I don't know about you, but I've been jolly busy recently, paddling up a creek without a paddle (or a creek), up to my armpits in alligators (I never metaphor I didn't like), and running around in ever-decreasing circles shouting, 'Don't Panic!' As a result, I haven't played with my prototype since last we spoke. Let's all power up our Arduinos and fire up our programs together. Are you ready? One… two… three… Go!

Hmm. Is your Serial Monitor displaying weird and wonderful characters something like the following?

```
7t)LT)L�ÕT)L�7t)L�#t...
```

If so, although this may accurately represent the date and time somewhere in the multi-universe, it fails to satisfy requirements here on planet Earth. When we see something like this, the first thing to do is check that the baud rate (think 'communication speed') in the Serial Monitor is set to the same value we've used in the call to the `Serial.begin()` function in our program. Sometimes the Arduino's integrated development environment (IDE) remembers the last value we used in the Serial Monitor… and sometimes it doesn't.

The Arduino uses a universal asynchronous receiver/transmitter (UART) function to communicate with the host computer. This is a very simple interface in which both the data format and transmission speeds are configurable. It requires only two wires to transmit and receive in both directions (both ends also require a connection to ground).

There are a variety of 'standard' baud rates used for serial communications implemented using UART protocols. Many of these are historical in nature. For example, 110 baud was used by ASR 33 Teletype machines (like the one in my office) to transfer data at 10 characters per second. The UART in the Arduino Uno supports values of 300, 600, 750, 1200, 2400, 4800, 9600, 19200, 31250, 38400, 57600 and more.

The default for the Arduino is 9600 baud. We can dictate the baud rate used by our program by specifying its value in the `Serial.begin()` function. And we can specify the value used by the Serial Monitor by clicking the down-pointing arrow next to the baud rate value in the interface (Fig.1).

In our previous column, we decided to speed things up by using 57600 baud. However, I commenced this current session by updating my IDE to the latest version. I'm assuming it was this that reset the Serial Monitor in my IDE back to 9600 baud. This resulted in a few seconds of gnashing of teeth and rending of garb before I realised what was going on.

As soon as I'd set the correct baud rate in my serial monitor, then I started to see the date and time values I was expecting, looking something like the following lines:

```
2024/2/10 (Saturday) 10:24:18
2024/2/10 (Saturday) 10:24:21
2024/2/10 (Saturday) 10:24:24
2024/2/10 (Saturday) 10:24:27
    :
   etc.
```

I feel like shouting, 'It's alive! It's alive!' Note that these values are updated every three seconds because we've included a three second (3000 millisecond) delay at the end of the `loop()` function in our program. (Try changing this to 1000 and observe that your display now updates each second.)

Drifting along

I'm sorry. I drifted off there for a moment. There's something hypnotic about watching this series of date and time values appearing on one's screen. But wait! Do the minutes values in your Serial Monitor fail to precisely match the time displayed in the menu bar of your host computer?

Don't freak out if there is a small discrepancy. One source of any potential difference is associated with the way in which we loaded the time and date into our RTC in the first place. The technique we used in our first program was to set the RTC to the date and time at which the program was *compiled*. This took place prior to uploading it into the Arduino and using it to initialise the RTC, all of which took time.

Another source of error is that RTCs aren't perfect and they're subject to drift. The DS3231 we're using is phenomenally accurate in the scheme of things. If you look at its data sheet – available at: **https://bit.ly/42CloQS** – you'll see this little scamp has a specified accuracy of ±2ppm (parts per million) over a temperature range of 0°C to +40°C. This means a worst-case drift of 2 seconds for every million seconds. Since there are 60 x 60 x 24 x 365 = 31,536,000 seconds in a year (assuming we aren't in a leap year), then this equates to a worst-case drift of ~0.2 seconds a day, ~1 second a week, or ~1 minute a year.

One final potential source of error may depend on whether you purchased a real DS3231 that was manufactured by Maxim Integrated, which is a subsidiary of the American multinational semiconductor company Analog Devices, or whether you acquired a cheaper Chinese 'knock-off,' which may or may not be up to par.

One of the things we will be talking about at some stage in the future is adding a few pushbutton switches to our prototype and using them to implement a menu system that will allow us to do things like setting the date and time by hand, but that's a discussion for another day because we currently have other *poisson à frire* (fish to fry).

One 7-segment display (sad face)

As you will doubtless recall, we currently have only a single 7-segment display at our disposal (sad face). This wasn't an issue with our first experiments in which we wished to display only the ten digits 0 through 9.

Later, when we came to experiment with our ultrasonic sensor (*PE*, February 2024), we decided to present distances on our display, where these distances were measured in centimetres using values between 0 and 99. As part of this, we elected to always employ two digits in the form 00, 01, 02… 97, 98, 99. As we said at that time: If we call our most-significant and least-significant digits MSD and LSD, respectively, and if we think of '…' as indicating a pause, then a first-pass view would be to display `MSD … LSD … … … MSD … LSD … … …`, and so forth.

Well, we now want to display the hours and minutes portions of the date and time values returned from our RTC. For our purposes, each of these values will involve two digits, which we can think of as HH:MM for hours and minutes. This means we have four digits to display. If we refer to the most-significant and least-significant hour and minute digits as H-MSD, H-LSD, M-MSD, and M-LSD, respectively, then one approach would be to display:

`H-MSD … H-LSD … … … M-MSD … M-LSD … … … … … … … …`
`… H-MSD … H-LSD … … … M-MSD … M-LSD –` and so forth.

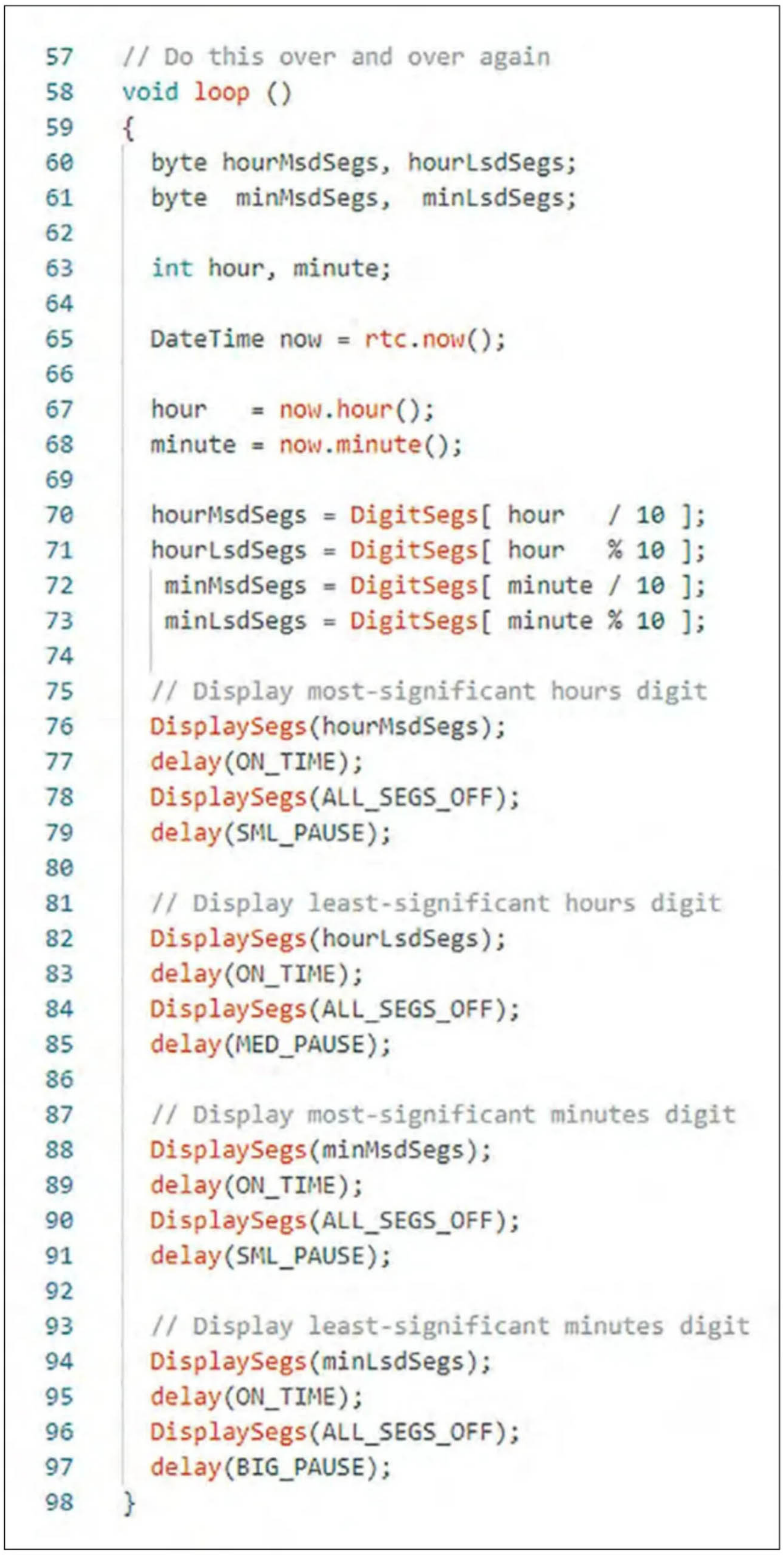
To put this another way, we will display the H-MSD and H-LSD digits with a short pause between them, wait for a medium-sized pause, display the M-MSD and M-LSD digits with a short pause between them, wait for a longer pause, and then do everything all over again, and again, and… you get the idea.

Decisions, decisions…

Now it's time for us to make a choice. Do we want to display time in a 24-hour or 12-hour clock format? We all use digital clocks in one form or another myriad times each day, typically without thinking about things too deeply, so I bet most people (not you, of course) think there are no surprises to be uncovered. Let's see, shall we?

In the case of a 24-hour clock, the day runs from midnight to midnight and is divided into 24 hours, which are numbered from 0 to 23. This scheme may be referred to as 'military time' in the US and parts of Canada, 'continental time' in the UK, and 'railway time' in other parts of the world.

By comparison, in the case of a 12-hour clock, the 24 hours of the day are divided into two 12-hour periods designated as a.m. (from the Latin *ante meridiem*, meaning 'before midday') and p.m. (from the Latin *post meridiem*, meaning 'after midday'). The 12 hours in each period are numbered: 12 (which, amusingly, acts as 0), 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11. Surprisingly, there is no widely accepted convention for how midday and midnight should be represented. However, in English-speaking countries, 12 a.m. means 12 o'clock midnight, while 12 p.m. indicates 12 o'clock noon.



```
// Do this over and over again
void loop ()
{
  byte hourMsdSegs, hourLsdSegs;
  byte  minMsdSegs,  minLsdSegs;

  int hour, minute;

  DateTime now = rtc.now();

  hour   = now.hour();
  minute = now.minute();

  hourMsdSegs = DigitSegs[ hour   / 10 ];
  hourLsdSegs = DigitSegs[ hour   % 10 ];
   minMsdSegs = DigitSegs[ minute / 10 ];
   minLsdSegs = DigitSegs[ minute % 10 ];

  // Display most-significant hours digit
  DisplaySegs(hourMsdSegs);
  delay(ON_TIME);
  DisplaySegs(ALL_SEGS_OFF);
  delay(SML_PAUSE);

  // Display least-significant hours digit
  DisplaySegs(hourLsdSegs);
  delay(ON_TIME);
  DisplaySegs(ALL_SEGS_OFF);
  delay(MED_PAUSE);

  // Display most-significant minutes digit
  DisplaySegs(minMsdSegs);
  delay(ON_TIME);
  DisplaySegs(ALL_SEGS_OFF);
  delay(SML_PAUSE);

  // Display least-significant minutes digit
  DisplaySegs(minLsdSegs);
  delay(ON_TIME);
  DisplaySegs(ALL_SEGS_OFF);
  delay(BIG_PAUSE);
}
```

Listing 3a. Displaying hours and minutes in 24-hour format.

Let's start by implementing the 24-hour clock system, which is the flavor our DS3231 RTC uses by default. In this case, each day officially begins in the middle of the night (a.k.a. midnight) at 00:00. A value of 12:00 occurs only at noon (ie, the middle of the day, a.k.a. midday). The last minute of the day begins at 23:59 and ends at 24:00, which is identical to 00:00 of the following day. This means our HH values will span 00 to 23, while our MM values will span 00 to 59.

24-hour clock

What we are going to do is to take the program that we discussed earlier (file **CB-Apr24-02.txt**) which displays the date and time from the RTC, plus take the program we used to display the 2-digit values from the ultrasonic sensor (file **CB-Feb24-03.txt** from *PE*, February 2024). Then we are going to munge and merge these little rascals together, add a sprinkle of programming magic, and 'Bob's your uncle' (or aunt, depending on your family dynamic).

I just did this and it works like a charm. But before you look at my solution, why don't you have a bash yourself? One 'word to the wise' is to remember to change the Arduino pin assignments to our `PinSegs[]` variable to be {9, 8, 7, 6, 5, 4, 3, 2} to match our new breadboard layout. We discussed this in our previous column, but it's easy to forget this sort of thing (as I just discovered to my cost).

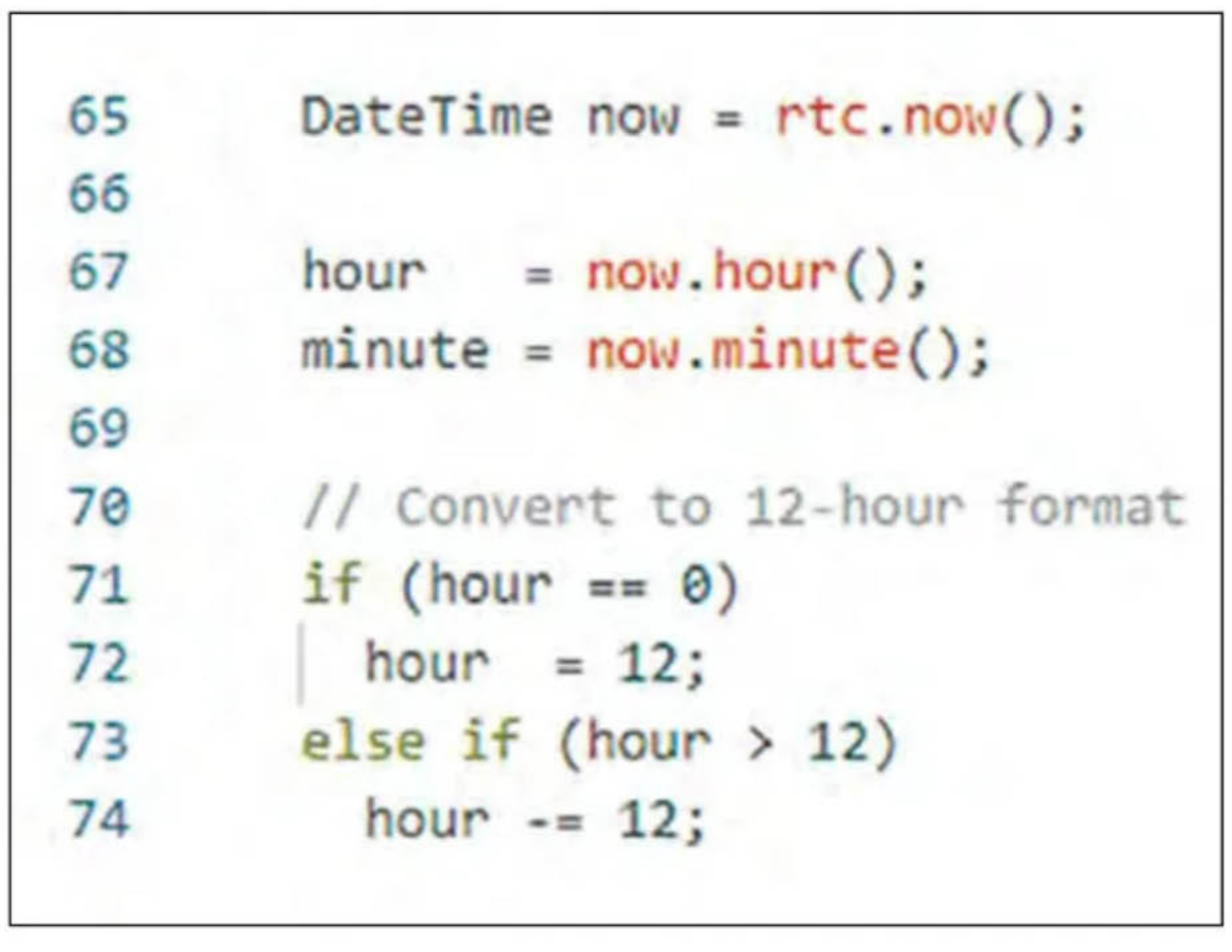
Once you have your own solution up and running, you can then compare it to mine (file **CB-Apr24-03.txt**). Since we've already seen most of this program before, we'll just look at the `loop()` function here (Listing 3a). Note that I've created three definitions, `SML_PAUSE`, `MED_PAUSE`, and `BIG_PAUSE` to which I've assigned values of 150, 450, and 1350 (milliseconds), respectively. I made each delay three-times the length of its predecessor because… that's just the sort of fellow I am. You can tweak these values to your heart's content to settle on what's most pleasing to your eye.

The first thing we do on Lines 60 and 61 is declare four variables of type `byte`. We're going to use these to store the segments associated with our H-MSD, H-LSD, M-MSD, and M-LSD digits. On Line 63 we declare two variables of type `int` (integer) called `hour` and `minute`. (I'll leave the mystery of what we are going to store in these to your imagination.)

As we discussed in last month's column, the RTC library we are using declares a new data type called `DateTime`. This is a structure containing multiple members whose values we can access using full-stop/period '.' delimiters (we introduced enumerated types, structures and type definitions in *PE*, December 2020). On Line 65, we declare a variable of type `DateTime` called `now`, to which we assign the values returned from a call to the `rtc.now()` function. On Lines 67 and 68 we copy the hours and minutes values out of our `now` variable into our `hours` and `minutes` variables (as we'll see, doing things this way will make our lives much easier when we come to implementing a 12-hour format).

As a reminder, our `DigitSegs[]` array contains the segments associated with the digits we wish to present on the 7-segment display. The way we've set things up, `DigitSegs[0]` contains the segments associated with the number 0, `DigitSegs[1]` contains the segments associated with the number 1, and so on.

Remember that, if we have a number between 0 and 99, then dividing it by 10 using the division operator '/' will return the most significant digit, while dividing it by 10 using the modulus operator '%' will return the least-significant digit. This is what we are doing on Lines 70 through 73 to determine which segments we wish to light up to display our most-significant and least-significant hour and minute values. After this, it's easy-peasy-lemon-squeezy to present the 4-digit sequence on our display, with each pair of digits being separated (ie, all segments turned off) by appropriate delays.



```
65      DateTime now = rtc.now();
66
67      hour   = now.hour();
68      minute = now.minute();
69
70      // Convert to 12-hour format
71      if (hour == 0)
72        hour  = 12;
73      else if (hour > 12)
74        hour -= 12;
```

Listing 4a. Converting to 12-hour format.

12-hour clock

A lot of people find it difficult to wrap their brains around the 24-hour clock format and they much prefer to use its 12-hour counterpart.

As fate would have it, the DS3231 can run in either 24-hour mode (the default) or 12-hour mode. If you look at the DS3231's data sheet, you will see that bit 6 of the hours register is defined as the 12-hour or 24-hour mode select bit. Alternatively, we can take the default 24-hour format returned from the DS3231 and convert it into a 12-hour form ourselves.

Which approach do you think we should use? I agree – let's do it ourselves. The easiest way to start out here is to take a step back and think about what we want to do (not surprisingly, this approach works well for most problems). The hours values returned by the DS3231 running in 24-hour mode are numbered 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23. If we wish to present time in a 12-hour format, we need to convert these values to be 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.

Well, that seems easy enough. If the hours value returned from the RTC is 0, we need to change our version to 12. Alternatively, if the hours value returned from the RTC is greater than 12, we need to subtract 12 from this value.

I just made this modification to my program (file **CB-Apr24-04.txt**). As we see in Listing 4a, the only difference between our previous program and our new version is the addition of a simple conversion routine on Lines 71 through 74.

Remember that the '-=' operator used on Line 74 is known as a compound operator. The following two lines have the same functionality:

```
hour -= 12;
hour = hour - 12;
```

We introduced the concept of compound operators quite some time ago (*PE*, March 2022). I know they can look a little 'tricky' when you are starting out. However, they do allow us to concisely capture what we are trying to do, and – after a while – you'll find it becomes easier to parse and process expressions using compound operators than their expanded versions. The more you capture code, the more you will grow to love compound operators (trust me, have I ever lied to you before?).

As one final thought before we proceed, if we are displaying time in a 12-hour format, we could use the decimal point (DP) segment on our 7-segment display to indicate if the time is a.m. (DP = Off) or p.m. (DP = ON). Personally, I'm usually aware of whether it's the morning, afternoon, or evening, so this would be superfluous to requirements. However, this may be of interest for those who are temporally challenged. We will return to consider this further (by which I mean using the DP to indicate a.m. or p.m.; not people who have a tortuous grasp on time) in our next column.

Two 7-segment displays (happy face)

I've said it before, and I'll say it again, when it comes to tedious, tiresome and turgid music, you'd have to go a long

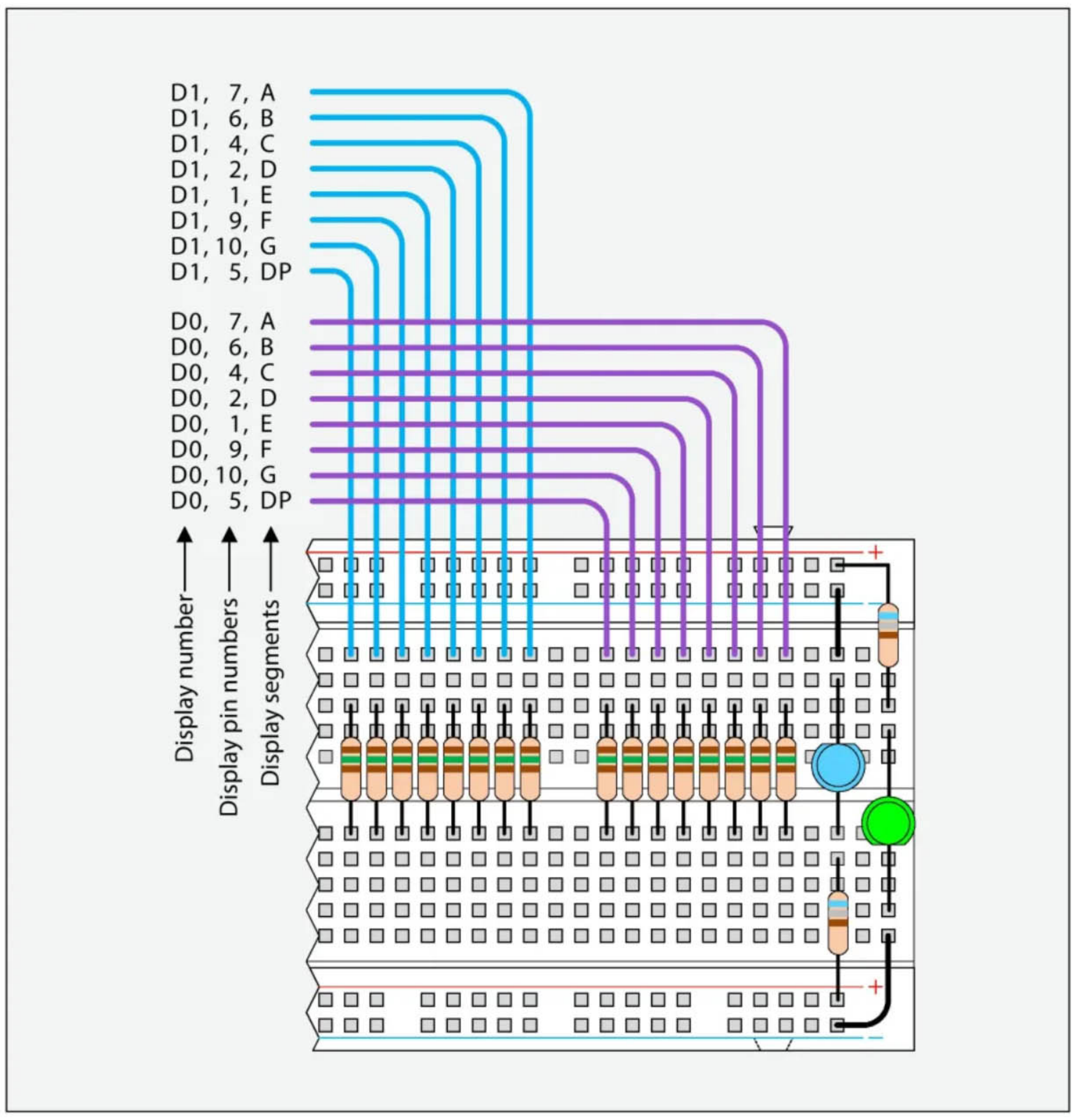


Fig.2. Adding eight more current-limiting resistors and wires.

way to find anything worse than *One* (commonly known for its opening line 'One is the loneliest number you'll ever do') as covered by Three Dog Night in 1969.

Defying all possibility, the second line is even worse than the first: 'Two can be as bad as one, it's the loneliest number since the number one' (I mean, come on, give me a break!). Well, we are here to refute the premise of this second line, because we are going to add a second 7-segment display to our breadboard (happy face).

Was that a squeaky squeal of excitement I just heard? (Yes, it was, but I just realised that it was me).

Adding more current-limiting resistors

The first thing I want you to do is return to the illustration showing our full prototype as it currently stands (file **CB-Apr24-01.pdf**).

In the following discussions, any references to wire colours reflect those I've used in my illustrations, your wire colours may vary depending on what you have available and what you decide to use.

We're going to leave the green and blue wires connecting the Arduino to our RTC on the lower breadboard as-is. In the case of the eight orange wires connecting the Arduino to one side of the 150Ω (brown-green-brown) current-limiting resistors on the upper breadboard, I want you to leave the ends plugged into the Arduino as-is, but unplug the ends connected to the breadboard.

In the fullness of time, we are going to deploy four 7-segment displays, which we will call D0, D1, D2 and D3. Currently, only D0 is installed on our breadboard. Our existing eight current-limiting resistors are connected to the eight segments on D0 by the eight purple wires.

Now, I want you to add eight more 150Ω resistors along with eight blue wires on the right-hand side of the upper breadboard, as illustrated in Fig.2.

Talking about the colour choices for our wires just reminded me of a time when not everyone had colour televisions and BBC snooker commentator Ted Lowe famously said something along the lines of: 'He's going for the pink, and for those of you watching in black and white, the pink is next to the green.' Hopefully you are not reading a black-and-white photocopy of this issue of *PE*.

Adding and connecting the second display

The next step is to add a second 7-segment display to the left-hand side of the upper breadboard, as illustrated in Fig.3. Once you've inserted this display, connect the blue wires coming from our new set of current-limiting resistors.

Now, this part is important. Observe the black wire shown as connecting pin 3 of display D0 to ground (0V). I want you to *remove* this wire but remember it for a thought experiment we are poised to perform. Similarly, observe the black wire shown as connecting pin 3 of display D1 to ground. *Do not* add this wire but – once again – remember it for the thought experiment to which we are racing (try not to lose your breath).

Wot? Not enough pins?

For our purposes here, we are going to ignore the fact that the Arduino Uno's six analogue pins A0 through A5 can also be used as digital input/output (I/O) pins. This is because we've been using our analogue pins for other tasks – like reading the value of a potentiometer (*PE*, November 2023) or reading the value from a light-dependent resistor (LDR) to measure the level of ambient light (*PE*, December 2023).

No one knows what the future holds, but there's more than a betting chance we might wish to add these types of functionalities into a future incarnation of our clock. Also, as discussed in our previous column, analogue pins A4 and A5 are no longer available to us because we are using them to provide I^{2}C communication with our RTC.

So, this is where we run into a problem. The Arduino Uno has only 14 digital input/output (I/O) pins numbered from 0 to 13. We can't use pins 0 and 1 (well, we can, but we're not going to) because these are used to implement any UART conversations with our host computer. Examples of these communications are when the host computer programs the Arduino, or when our programs send messages to the host computer using `Serial.print()` function calls.

This means we have only 12 digital I/O pins at our disposal. Thus far, we've been using eight of these pins (2 through 9) to drive the eight segments on our 7-segment display (remember, the decimal point counts as a segment in its own right). This means we have only four digital I/O pins (10 through 13) remaining that are available to us.

But wait, there's more. Although we aren't doing so presently, in the past we've used our four free digital pins

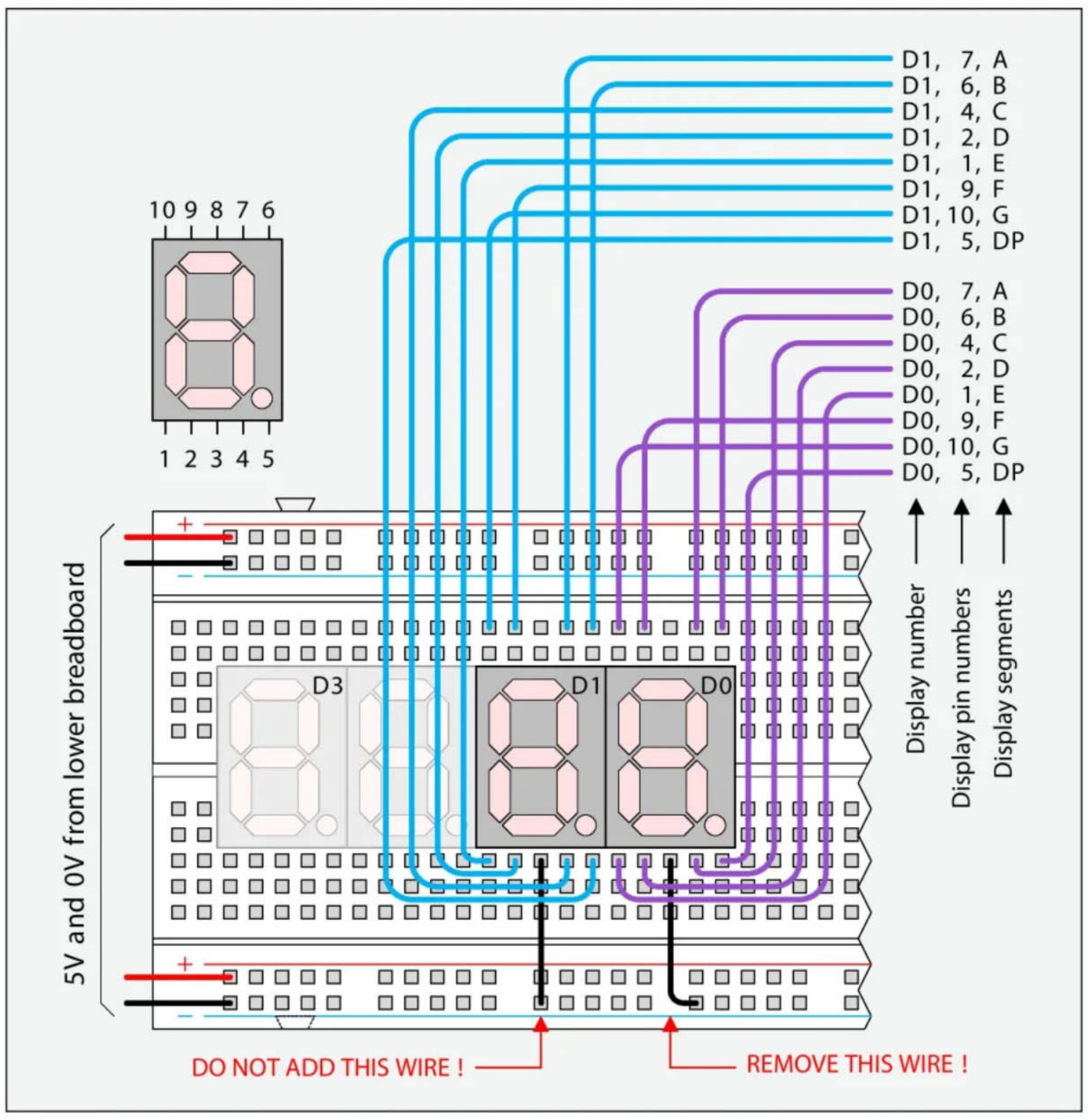


Fig.3. Adding and connecting the second 7-segment display (D1).

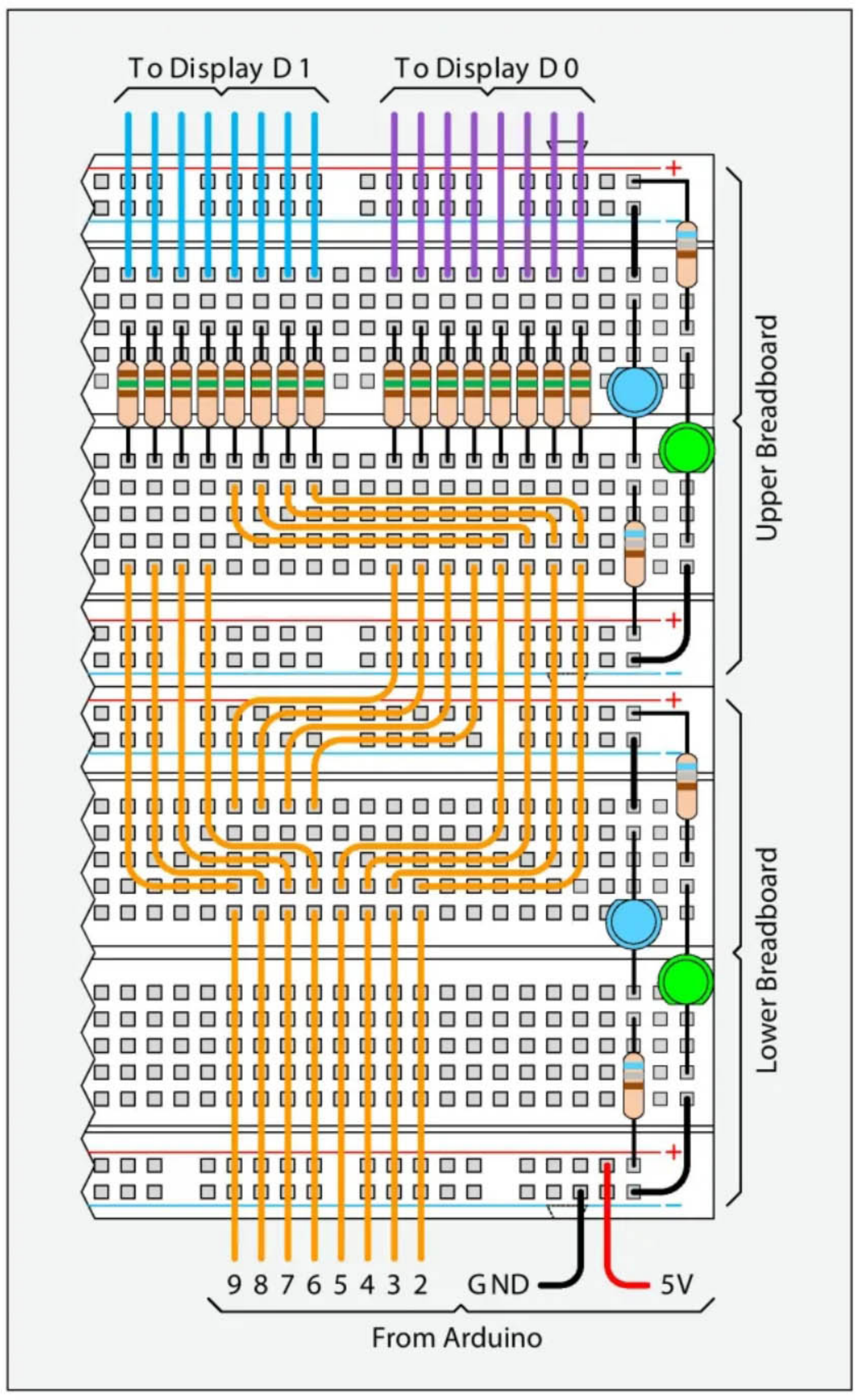


Fig.4. Reconnecting the Arduino.

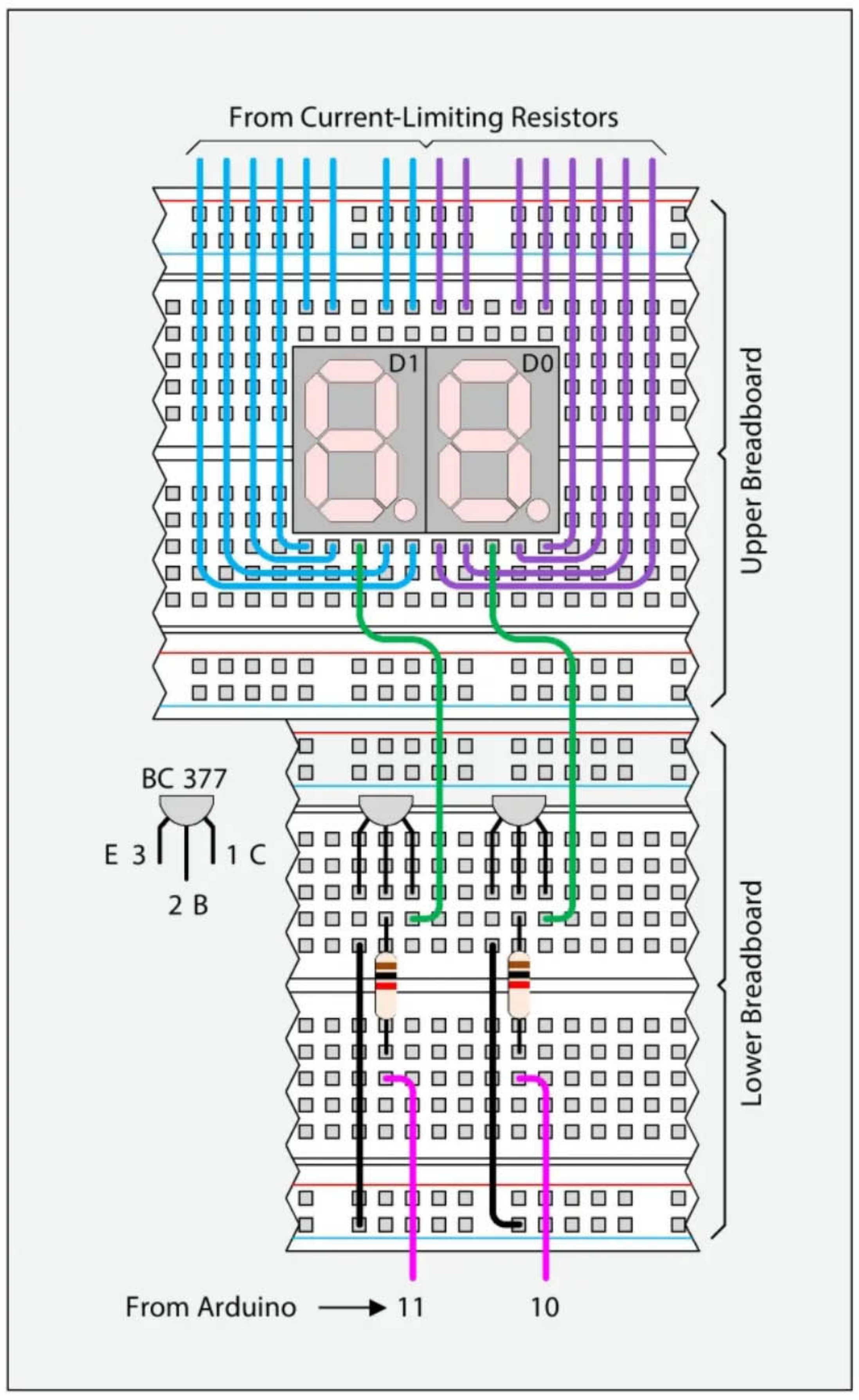


Fig.5. Adding two transistors.

to implement tasks like driving our piezoelectric buzzer (*PE*, December 2023 and January 2024) and communicating with our ultrasonic sensor (*PE*, February 2024). Once again, we may wish to add these types of functionalities into a future incarnation of our clock.

So, ignoring any potential additional functionalities for the moment, now that we've added our second display, we have 16 segments to drive and only 12 pins to drive them with, which presents us with a bit of a poser…

Reconnecting the Arduino

As a first step, I want you to connect the free ends of the orange wires coming from the Arduino to free rows of terminal strips on the lower breadboard. Next, I want you to add more orange wires such that each signal from the Arduino drives the same segment on both displays. For example, we want pin 2 on the Arduino to be connected to the free ends of the current-limiting resistors driving the A segments on both the D0 and D1 displays, pin 3 on the Arduino to be connected to the free ends of the current-limiting resistors driving the B segments on both the D0 and D1 displays, and so forth.

Online resources

For the purposes of this series, I'm going to assume that you are already familiar with fundamental concepts like voltage, current and resistance. If not, you might want to start by perusing and pondering a short series of articles I penned on these very topics – see: **https://bit.ly/3EguiJh**

Similarly, I'll assume you are no stranger to solderless breadboards. Having said this, even if you've used these little scamps before, there are some aspects to them that can trap the unwary, so may I suggest you feast your orbs on a column I wrote just for you – see: **https://bit.ly/3NZ70uF**

Last, but not least, you will find a treasure trove of resources at the **Arduino.cc** website, including example programs and reference documentation.

This is easy to do in practice (it's also easy to mess up if you aren't paying attention), but it's @#$%^ difficult to draw as – once again – I just discovered to my cost. The thing is that, to make things understandable in my illustrations, I don't want any of these wires to cross each other.

When I was a kid, my dad drew three boxes on a piece of paper and said they represented three houses. Then he drew three circles on the same piece of paper and said they represented three utility companies (gas, water and electricity). He told me that my goal was to use lines to connect each house with each utility company *without any of the lines crossing over*. He promised me this was possible (I'm sure he had his fingers crossed behind his back when he said this). This kept me occupied for days, so – as far as dad was concerned – it was 'mission accomplished.'

Just for giggles and grins, I created a drawing showing the resistors in the upper breadboard and the wires from the Arduino plugged into the lower breadboard – see file **CB-Apr24-05.pdf**. Why don't you download this, print it out, and try adding the requisite wires without any of them crossing. Once you've finished, compare your version to mine (Fig.4).

Yes, I'm afraid that is a somewhat self-satisfied smirk you see plastered on my visage. I must admit that I'm rather proud of this implementation.

Components from Part 1

LEDs (assorted colours)	**https://amzn.to/3E7VAQE**
Resistors (assorted values)	**https://amzn.to/3O4RvBt**
Solderless breadboard	**https://amzn.to/3O2L3e8**
Multicore jumper wires (male-male)	**https://amzn.to/3O4hnxk**

Components from Part 2

7-segment display(s)	**https://amzn.to/3Afm8yu**

Components from Part 5

Momentary pushbutton switches	**https://amzn.to/3Tk7Q87**

Components from Part 6

Passive piezoelectric buzzer	**https://amzn.to/3KmxjcX**

Components for Part 9

SW-18010P vibration switch	**https://bit.ly/46SfDA4**

Components for Part 10

Breadboard mounting trimpots	**https://bit.ly/3QAuz04**

Components for Part 12

Light-Dependent Resistor	**https://bit.ly/3S2430m**

Components for Part 13

BC337 NPN Transistor	**https://bit.ly/40xAgyS**

Components for Part 14

HC-SR04 Ultrasonic Sensor	**https://bit.ly/49AMBq4**

Components for Part 15

Real-Time Clock (RTC)	**https://bit.ly/3S9OjHl**

What would happen if…

This is where we arrive at the thought experiment we discussed earlier. Suppose our displays were to be wired up as shown in Fig.3. That is, *with* the two black wires connecting pin 3 on each display to ground. What would happen if we started to run one of our old programs that repeatedly counts from 0 to 9 while driving the corresponding segments on the 7-segment displays?

Well, two things would occur. First, since our two displays are wired in parallel (side by side), the same value would be presented on each display, which isn't tremendously useful to us in the scheme of things. Second, since we are using 150Ω current-limiting resistors, each segment will be drawing 20mA of current when it's turned on (we discussed this in excruciating detail in *PE*, February 2023). As we now have two displays in parallel, this means the Arduino pins driving our displays will have to source 2 x 20mA = 40mA of current. They can do this… *but not for long…* especially if a lot of them are doing it at the same time.

One solution to our 40mA problem would be to double the values of our current-limiting resistors to 300Ω, thereby halving the current flowing through each segment to 10mA. However, in addition to reducing the brightness of the displays, this wouldn't help us with the fact that both displays are presenting the same values. Another, much more promising solution is…

Adding two transistors

Do you remember when we used a BC377 transistor to vary the brightness of a single 7-segment display (*PE*, January 2024)? In that case, we turned the transistor on and off very quickly, and we varied the ratio between it being on and off to vary the brightness of the display.

Well, we are going to do something similar here. In this case, however, we are going to use two transistors to ensure that only one display is on at any time. In addition to limiting the current to a single display at a time, this will also allow us to present different values on each display. So, after first ensuring that there are no black wires connecting pin 3 on each display to ground (Fig.3), we are going to add two BC377 transistors, two resistors and two control wires coming from the Arduino (Fig.5).

For your delectation and delight, you can download an image of our latest and greatest hardware setup including all the changes we've discussed in this column (file **CB-Apr24-06.pdf**).

Next time

In our next column, we are going to use our dual displays to first display the hours (HH) and then the minutes (MM), after which we will start to consider how we might further boost our clock to employ four 7-segment displays (be still my beating heart)! In the meantime, I think it would be a good idea for you to start performing your own experiments with our current dual display setup – *just make sure that you only ever have one of our transistors turned on at a time!*

Cool bean Max Maxfield (Hawaiian shirt, on the right) is emperor of all he surveys at **CliveMaxfield.com** – the go-to site for the latest and greatest in technological geekdom.

Comments or questions? Email Max at: **max@CliveMaxfield.com**

Teach-In 8 CD-ROM Exploring the Arduino

This CD-ROM version of the exciting and popular *Teach-In 8* series has been designed for electronics enthusiasts who want to get to grips with the inexpensive, immensely popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. *Teach-In 8* provides a one-stop source of ideas and practical information.

The Arduino offers a remarkably effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle wirelessly or via the Internet. *Teach-In 8* is based around a series of practical projects with plenty of information for customisation. The projects can be combined together in many different ways in order to build more complex systems that can be used to solve a wide variety of home automation and environmental monitoring problems. The series includes topics such as RF technology, wireless networking and remote web access.

FREE CD-ROM
ELECTRONICS TEACH-IN 8
£8.99
FREE CD-ROM SOFTWARE FOR THE TEACH-IN 8 SERIES
INTRODUCING THE ARDUINO
Teach-In 8
PLUS...

PRICE £8.99

Includes P&P to UK if ordered direct from us

PLUS: PICs and the PICkit 3 – A beginners guide

The CD-ROM also includes a bonus – an extra 12-part series based around the popular PIC microcontroller, explaining how to build PIC-based systems.

SOFTWARE

The CD-ROM contains the software for both the *Teach-In 8* and PICkit 3 series.

ORDER YOUR COPY TODAY!
JUST CALL 01202 880299 OR VISIT www.epemag.com

AUDIO OUT

By Jake Rothman

Back to the buffers – Part 3

Following the stripboard designs for a discrete buffer in last month's *Audio Out* I have produced a PCB version; this board will be available from the *PE PCB Service* from *next* month. All the versions of the circuit given in the last issue can be built on this PCB, but it is most likely to be used for the high-voltage Darlington version given originally in Fig.14 (shown again here in Fig.24 in case you missed it). Unlike the discrete op amp PCB, the components are more spaced-out, reflecting the simplicity of the circuit. This helps with experimentation, allows for big audiophile capacitors and is shown in Fig.25. The single-rail version is shown in Fig.26.

I've also had a few more useful buffering circuit design ideas, which will be the topic of this and next month's pieces.

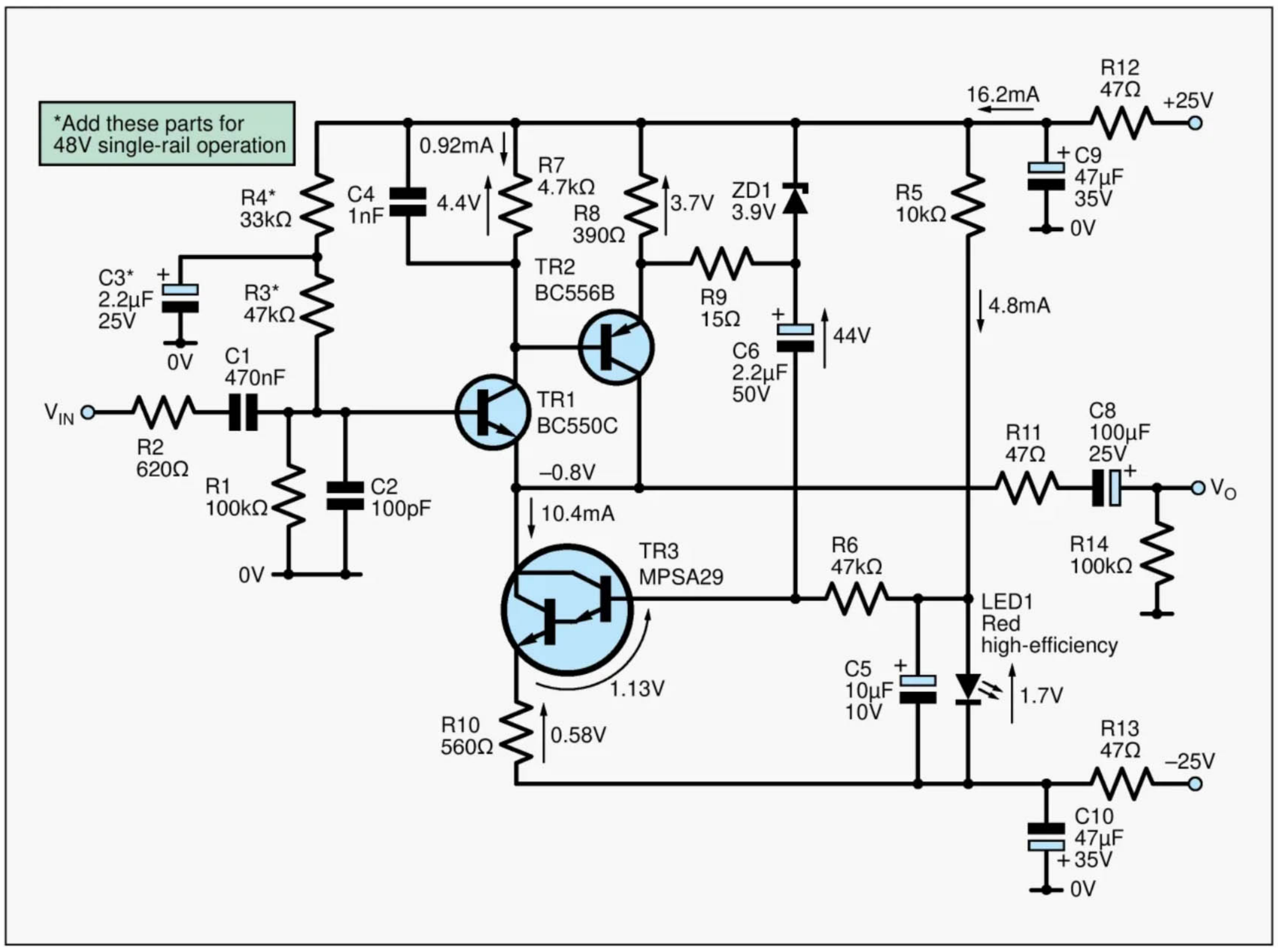

*Add these parts for 48V single-rail operation
R12 47Ω
16.2mA
+25V
0.92mA
R4* 33kΩ
C4 1nF
4.4V
R7 4.7kΩ
R8 390Ω
3.7V
ZD1 3.9V
R5 10kΩ
C9 47µF 35V
0V
C3* 2.2µF 25V
R3* 47kΩ
0V
TR2 BC556B
R9 15Ω
C6 2.2µF 50V
44V
4.8mA
V_IN
R2 620Ω
C1 470nF
TR1 BC550C
R1 100kΩ
C2 100pF
0V
–0.8V
10.4mA
R11 47Ω
C8 100µF 25V
V_O
R14 100kΩ
TR3 MPSA29
R6 47kΩ
LED1 Red high-efficiency
1.13V
C5 10µF 10V
1.7V
R10 560Ω
0.58V
R13 47Ω
–25V
C10 47µF 35V
0V

Fig.24. Buffer circuit repeated from Fig.14 in *Part Two* (*PE*, March 2024).

JFET input buffer

I'm sorry, but I just can't leave anything alone! A new version of the JFET buffer circuit (originally Fig.22) is shown in Fig.27. This 're-optimisation' often happens when I receive a PCB because it's easier to change parts and test compared to original prototype stripboard versions. I found I got better results with the Fairchild/OnSemi J113 JFET than the BF244A. They are cheaper as well, around 13p each (when buying 100 from Mouser). The distortion at $1V_{rms}$ into 600Ω was 0.0008% with R8 changed to 300Ω. (At $4V_{rms}$ it was 0.006%, and 0.3% at $6V_{rms}$). Playing with this design reminded me that one of the best things about discrete circuits is the way the resistor values can be optimised for any given operating condition.

I also increased the rails to ±18V, the maximum allowable, since the JFET has a voltage limit of 35V V_{ds} (drain-source voltage). Of course, I then had to change all the resistor values. Note that some components are omitted, such as the single-rail biasing parts, R3, R4 and C3. I also removed the anti-spike components R9 and ZD1. This was done because the later stages in most audio systems clip first. To run the buffer at ±25V, (matching the discrete op amp power rail) the decoupling resistors R12 and R13 can be increased to 470Ω.

I was surprised to see noise curves for the J111 to J113 JFETs on the OnSemi data sheet, I always thought these devices were intended for switching rather than audio designs. At an operating current of

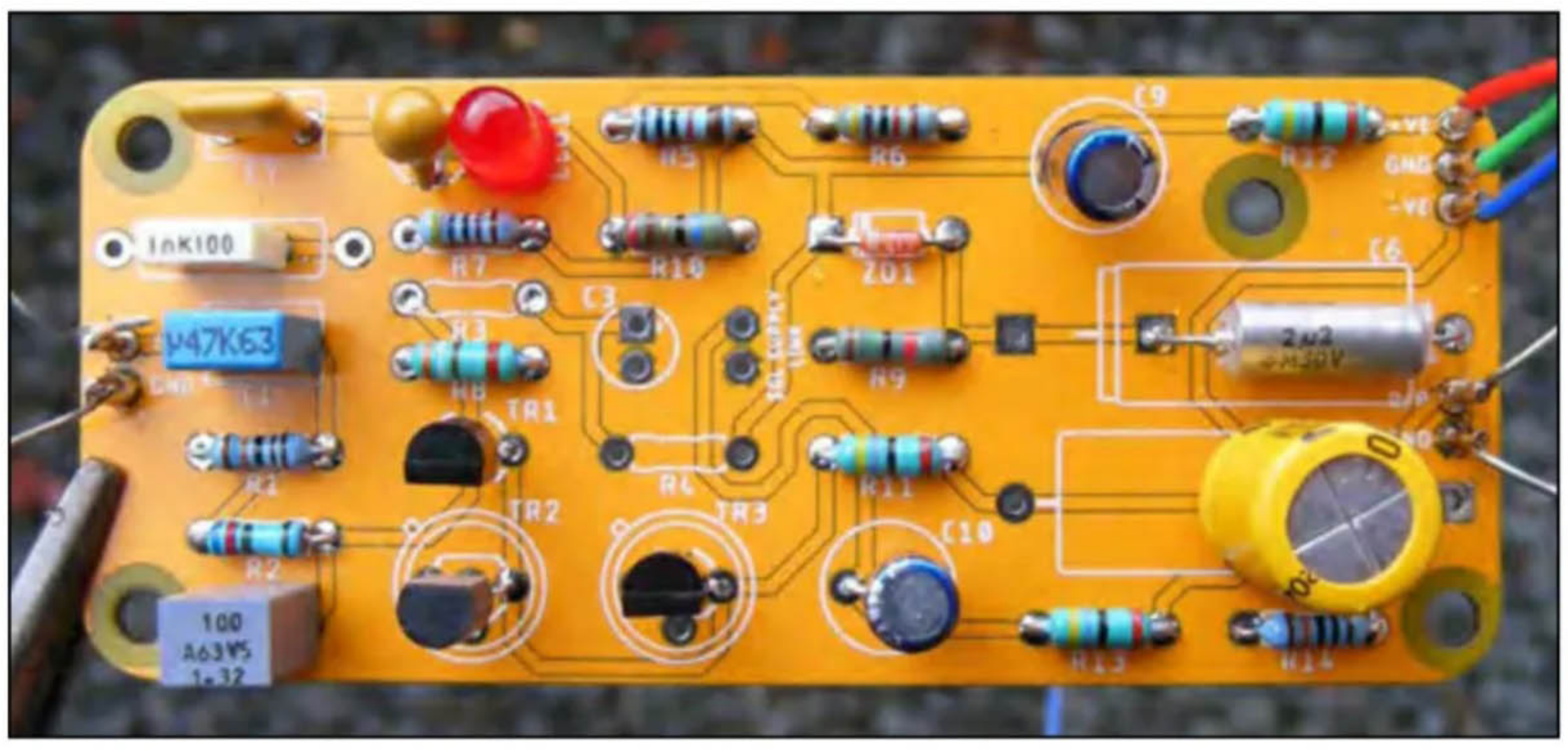


Fig.25. The assembled discrete buffer amplifier.

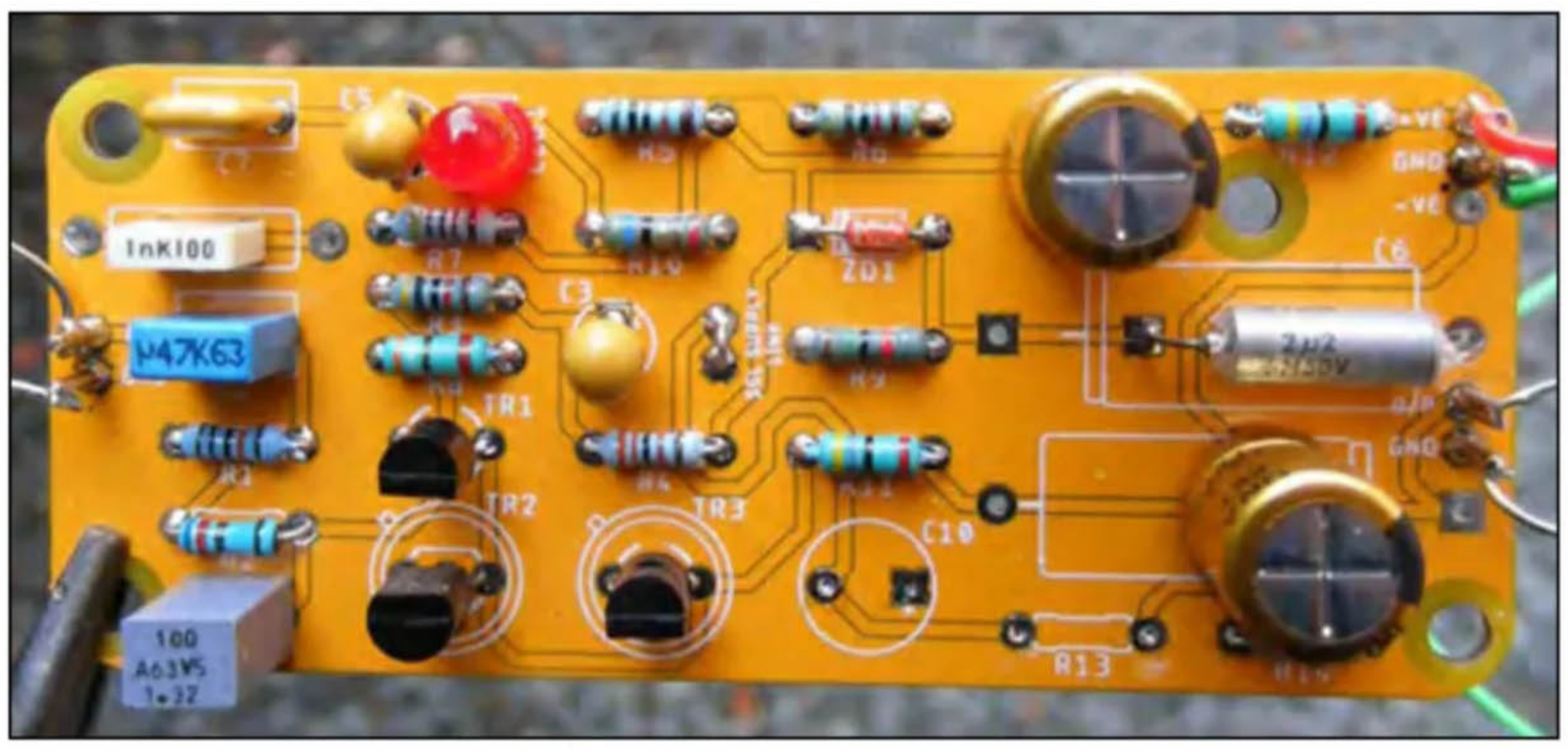


Fig.26. Single-rail version of discrete buffer amplifier. Extra biasing parts: R3, R4, C3 and SGL supply link installed. C10 is omitted.

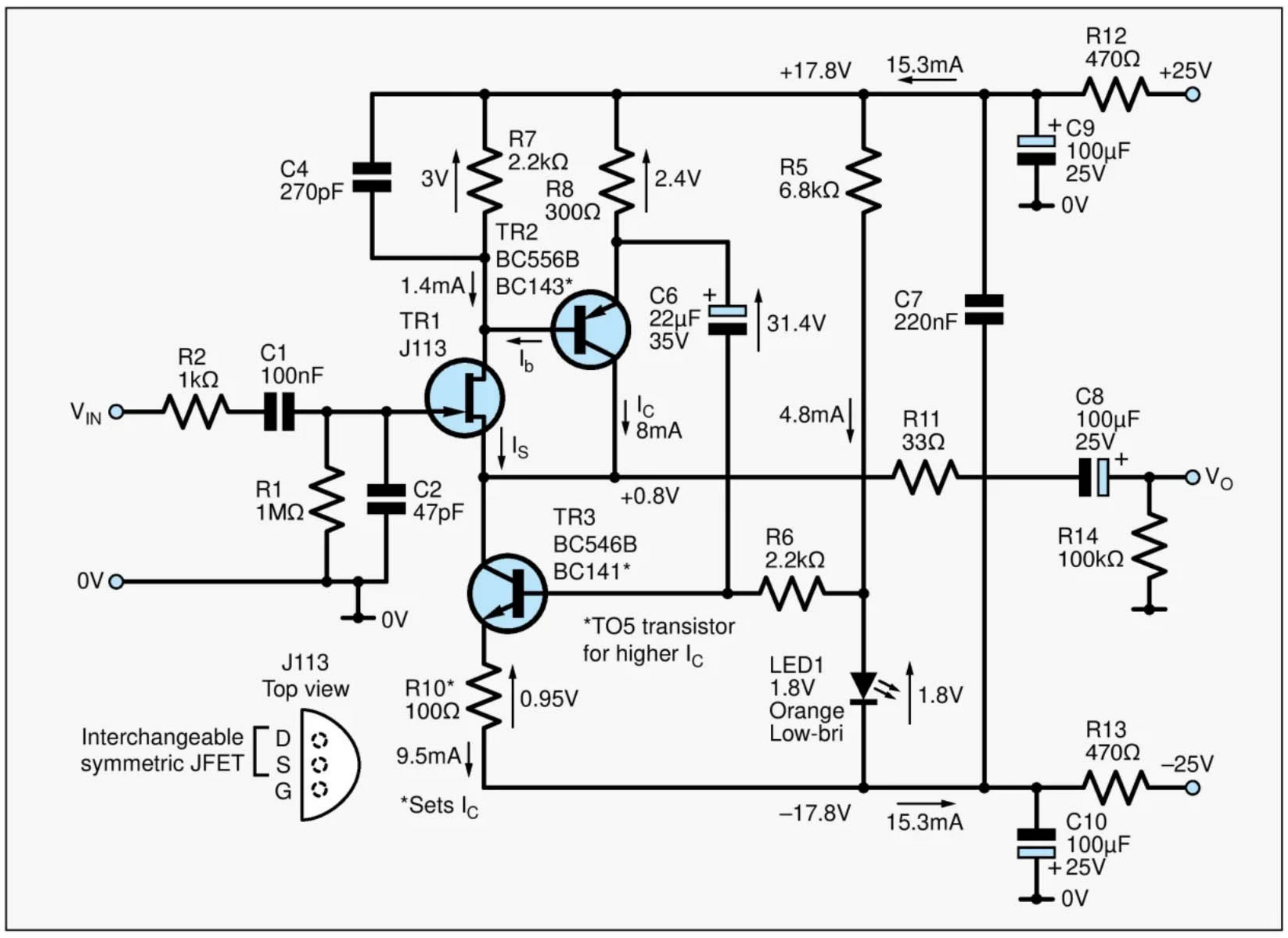

R12
470Ω
+25V
+17.8V
15.3mA
C9
100µF
25V
0V
C4
270pF
3V
R7
2.2kΩ
R8
300Ω
2.4V
R5
6.8kΩ
TR2
BC556B
BC143*
1.4mA
TR1
J113
C6
22µF
35V
31.4V
C7
220nF
R2
1kΩ
C1
100nF
Ib
VIN
IC
8mA
4.8mA
R11
33Ω
C8
100µF
25V
VO
IS
R1
1MΩ
C2
47pF
+0.8V
TR3
BC546B
BC141*
R6
2.2kΩ
R14
100kΩ
0V
0V
*TO5 transistor
for higher IC
J113
Top view
Interchangeable
symmetric JFET
D
S
G
R10*
100Ω
0.95V
9.5mA
*Sets IC
LED1
1.8V
Orange
Low-bri
1.8V
R13
470Ω
–25V
–17.8V
15.3mA
C10
100µF
25V
0V

Fig.27. JFET buffer circuit – the supply rails are ±18V and resistors R12 and R13 allow it to be used on ±25V supplies. (Overlay diagram will be provided next month.)

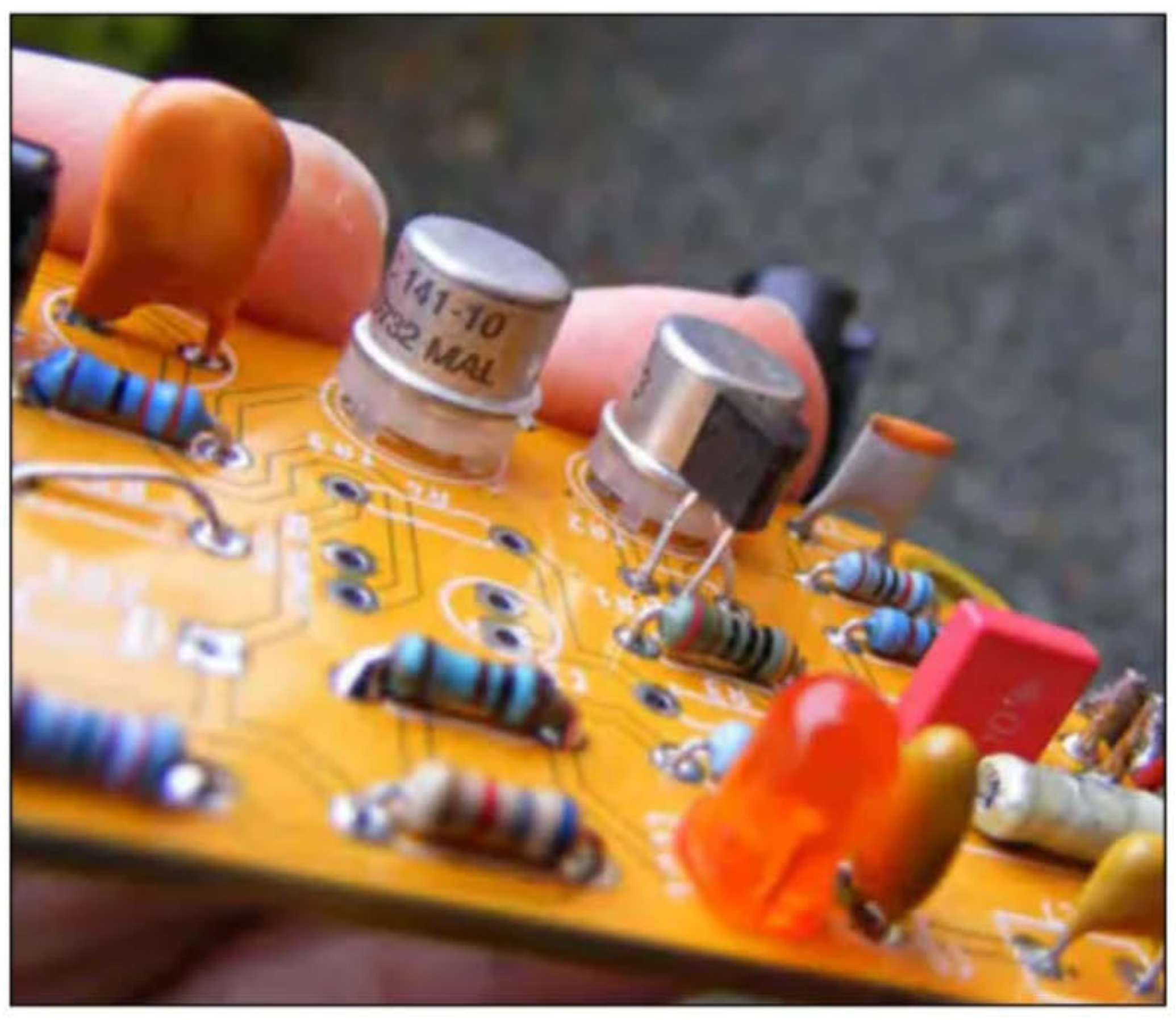


Fig.28. JFET buffer amplifier PCB. Note the cross-legged mounting of JFET TR1 and that the output transistors TR2 and TR3 have been upgraded to TO5 devices.

1.35mA, the noise voltage of the JFET is 2.5nV/√Hz, better than the NE5534 and equal to most expensive audio JFET op amps. By increasing the current up to 10mA, the noise can be reduced to just 1.3nV/√Hz, not bad at all.

JFET buffer assembly

The JFET buffer assembled on the PCB is shown in Fig.28. Note the link for R9 and the omitted components. Sadly, the pinout of the J113 JFET was different, not centre-gate, so a 'leg wiggle' was needed. Note also that the board can accommodate TO5-cased transistors for TR2 and TR3, as shown. This allows a higher output stage current if desired. It's also much easier to fit heatsink clips if needed.

JFET buffer component list

Semiconductors

TR1	J113 N-channel JFET
TR2	BC556 PNP small-signal bipolar
TR3	BC546 NPN small-signal bipolar
LED1	orange 3mm (or similar with a V_f of 1.8V at 5mA)
ZD1	omitted

Capacitors

C1	100nF 5mm polyester 10%
C2	47pF 2.5mm ceramic 10% NP0
C3	omitted
C4	270pF 2.5mm ceramic 10% NP0
C5	10µF 10V 2.5mm pitch radial electrolytic
C6	22µF 35V 2.5mm pitch radial electrolytic
C7	220nF 2.5mm ceramic 20% X7R
C8	100µF 35V 5mm pitch radial electrolytic, non-polarised preferred. Nichicon UEP1E101MPD from Mouser.
C9, C10	100µF 25V 3mm pitch two off

Resistors

R1	1MΩ
R2	1kΩ
R3, R4	not used
R5	6.8kΩ
R6, R7	2.2kΩ
R8	300Ω
R9	link
R10	100Ω
R11	33Ω
R12, R13	470Ω
R14	100Ω

Balanced buffer

Now that we have a good discrete op amp and buffer circuit, we can start combining them into useful audio systems. The classic use of a buffer in Hi-Fi pre-amplifiers is to isolate the low and changing input impedance of a Baxandall tone control from the volume control, as shown in Fig.29.

I mentioned in the last issue that another pro audio use is to buffer the inputs of a differential op amp circuit to provide equal high input impedance for good CMRR. This is shown in Fig.30. The op amp resistors have to be kept low for low noise, resulting in a difficult-to-drive low input impedance that needs buffering. Another problem is that the input resistances of the differential op amp circuit are unequal if they are the same value resistors on both inputs. I changed the values to equalise this, giving equal loading to the buffers. This ensures the even-order distortion harmonics produced by the buffers are equal. These harmonics cancel out in

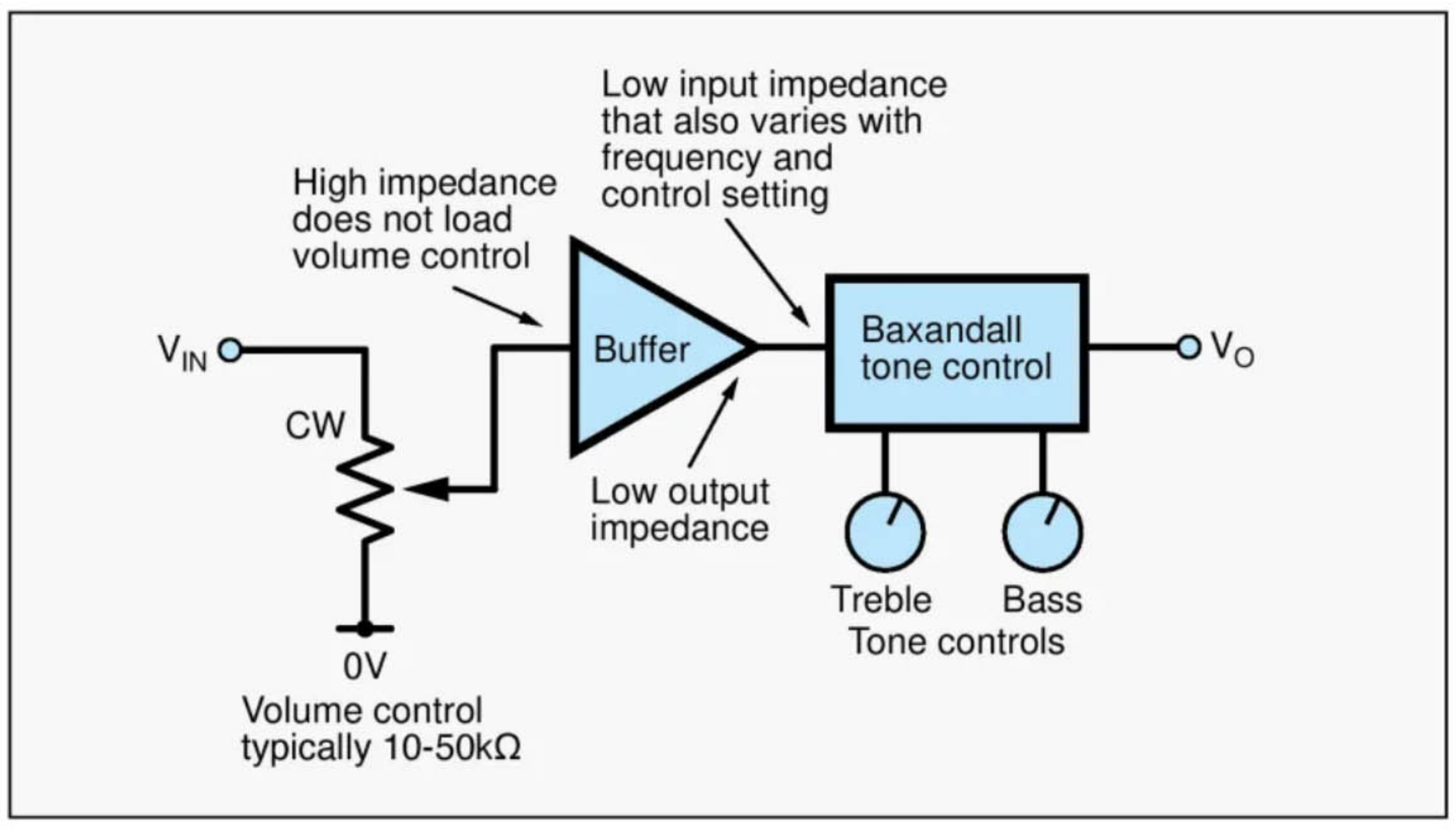

Low input impedance
that also varies with
frequency and
control setting
High impedance
does not load
volume control
VIN
CW
Buffer
Baxandall
tone control
VO
Low output
impedance
0V
Volume control
typically 10-50kΩ
Treble
Bass
Tone controls

Fig.29. A common use of a buffer in audio is driving a filter network such as a Baxandall tone control. Filters need to be driven from a low impedance and log volume potentiometers need to be loaded by a high impedance.

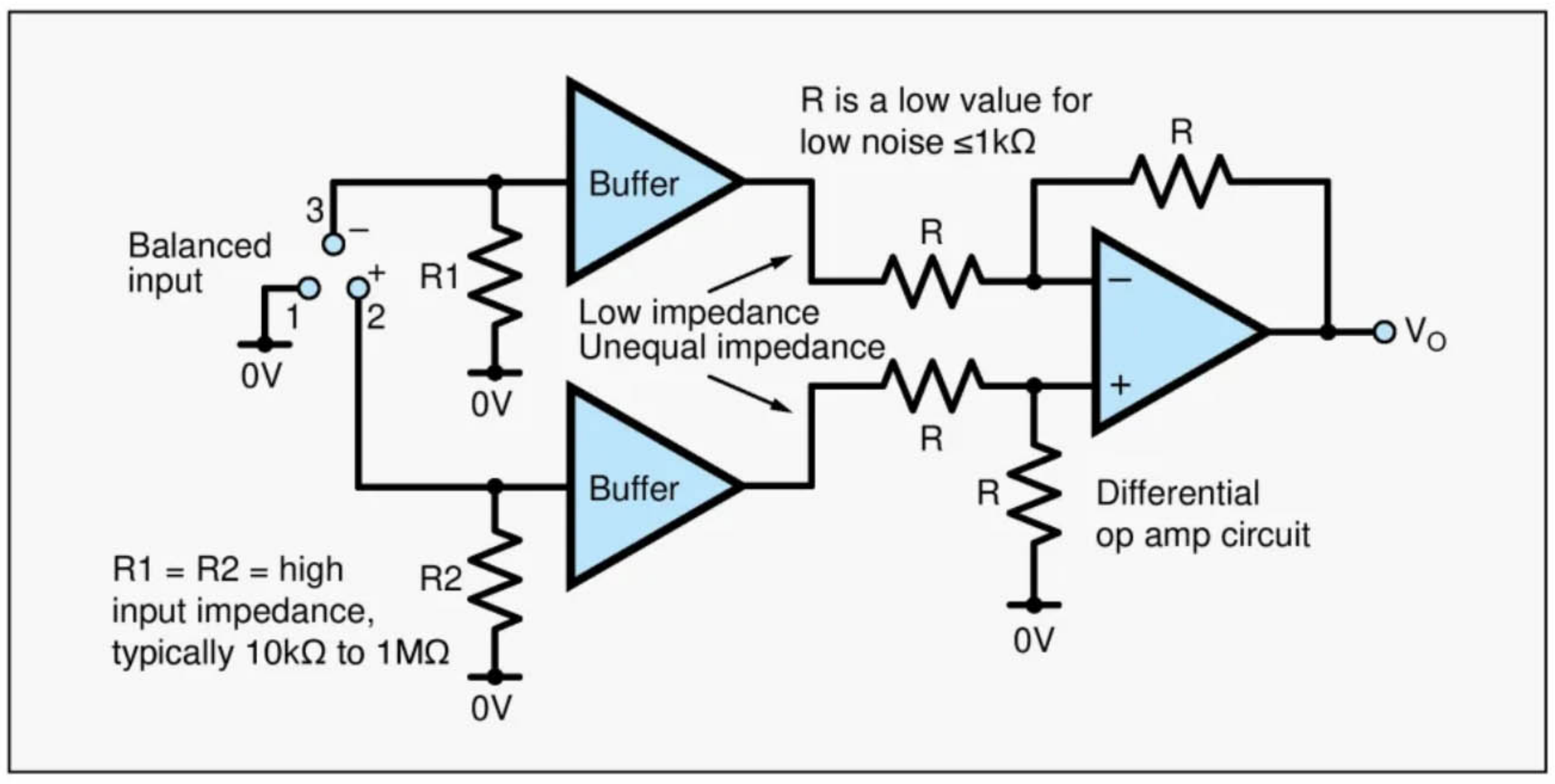

R is a low value for
low noise ≤1kΩ
Buffer
R
Balanced
input
3
–
+
1
2
R1
0V
0V
R
Low impedance
Unequal impedance
–
+
VO
Buffer
R
R
Differential
op amp circuit
R1 = R2 = high
input impedance,
typically 10kΩ to 1MΩ
R2
0V
0V

Fig.30. Instrumentation amplifier arrangement of a buffered differential amplifier. This gives inputs that are high impedance and equal impedance. The buffers also allow the resistors to have low values, minimising noise.

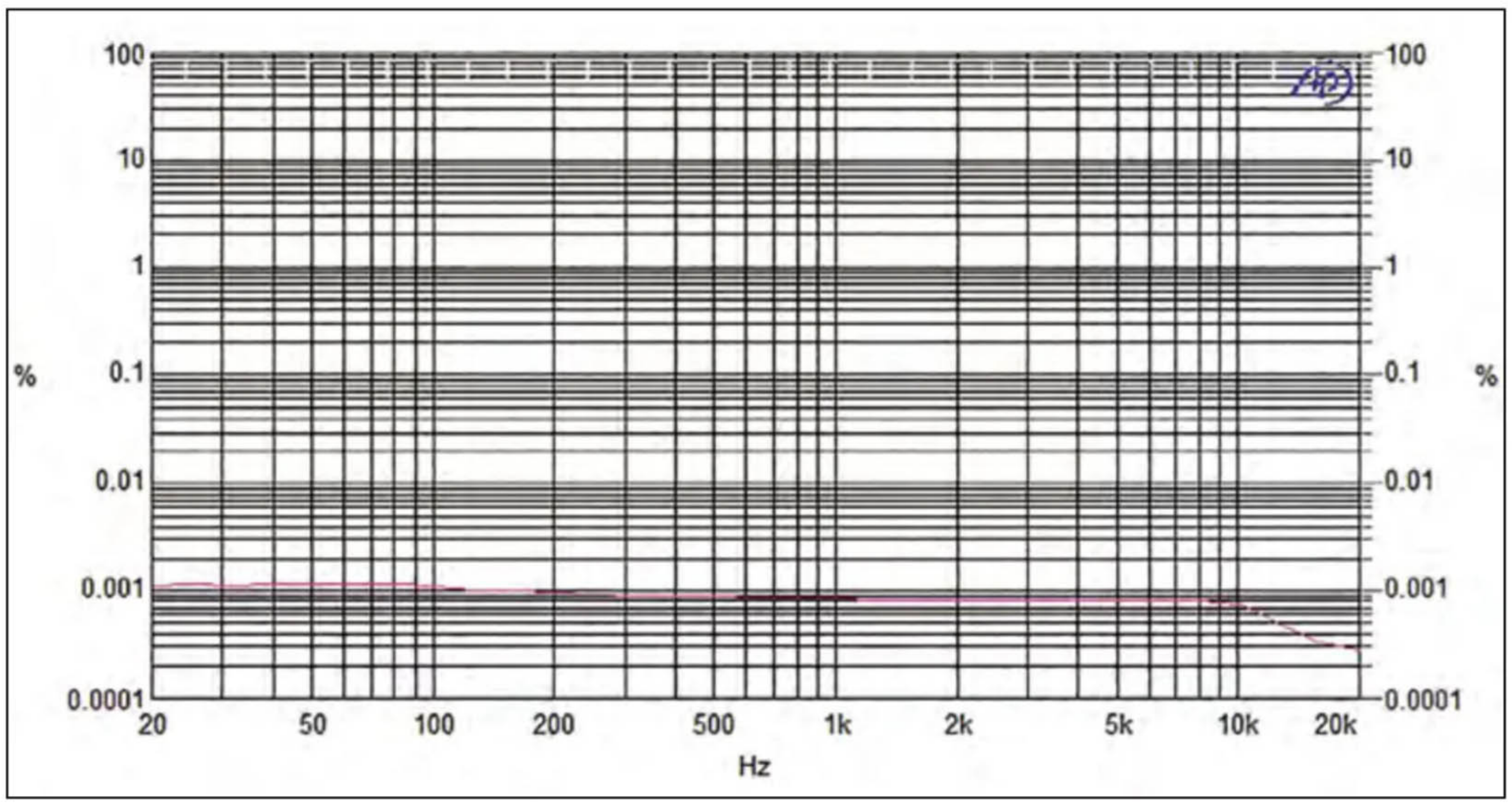


Fig.31. Distortion curve of differential amplifier with two JFET buffers on the inputs. Input 1Vrms output 3.2V$_{rms}$ (9.1V$_{pk-pk}$) into 600Ω.

the differential op amp. I also decided to add a CMRR trimmer (PR3) in the network. Looking at the final distortion curves for the whole system in Fig.31, I think all this fiddly tweaking and experimenting was worth it. The cancellation effect achieved was 8dB reduction on the second harmonic.

Negative resistance?

While making these changes I was surprised to find that the input resistance on the inverting input of the op amp was *less* than the input resistor of 1kΩ. How could this be? The answer was that the same voltage present on the non-inverting pin of the op amp was also present on the inverting pin, which is always the case with linear op amp circuits with negative feedback. This voltage is 180° out of phase compared to the inverting input terminal voltage. This increases the overall voltage across the resistor, giving it effectively a lower value (the opposite of bootstrapping) as shown in Fig.32. I couldn't find this effect in differential amplifiers mentioned in any textbook, but like many odd things in analogue design, it's still just Ohm's law.

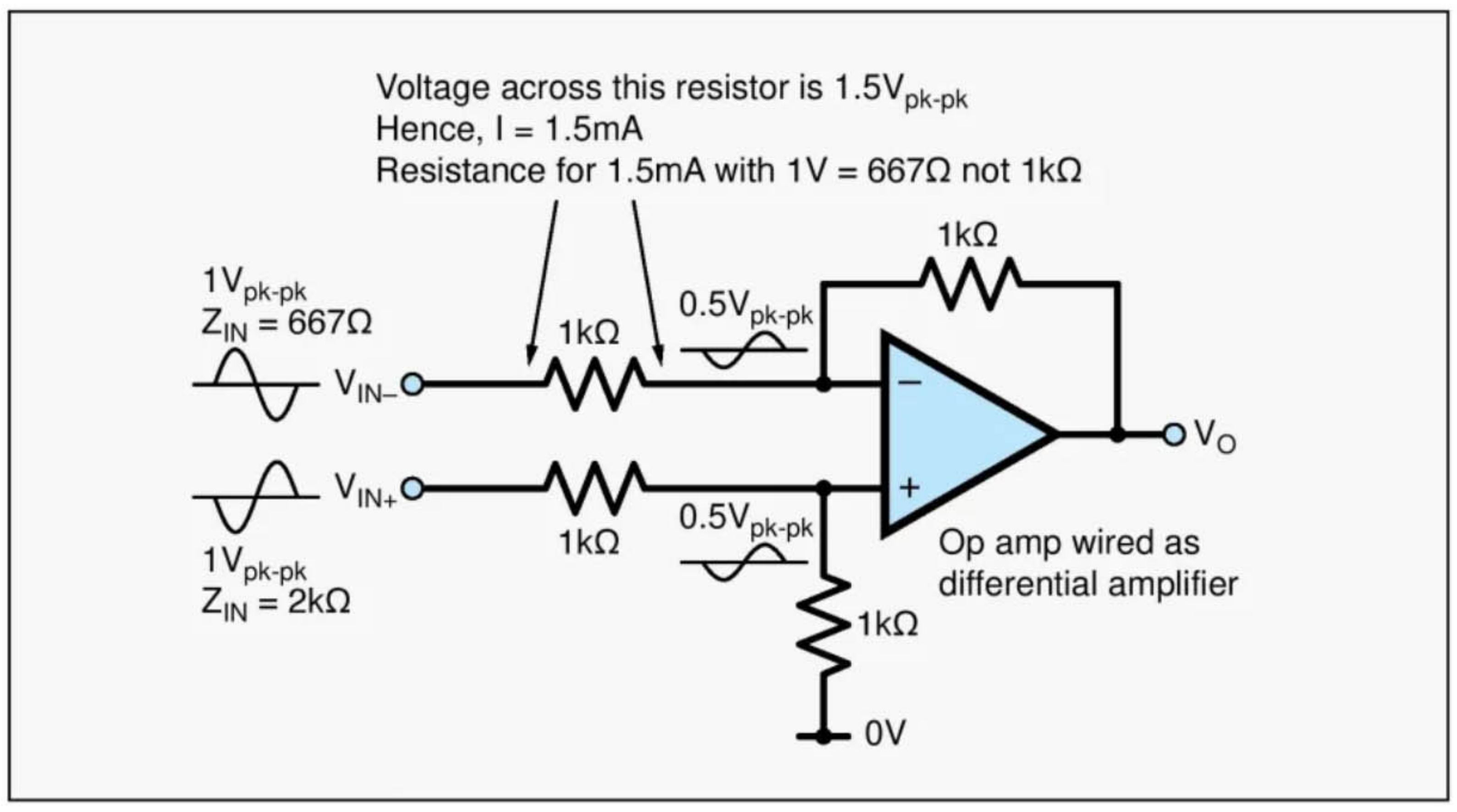

Voltage across this resistor is 1.5Vpk-pk
Hence, I = 1.5mA
Resistance for 1.5mA with 1V = 667Ω not 1kΩ
1Vpk-pk
ZIN = 667Ω
VIN–
1kΩ
0.5Vpk-pk
1kΩ
VO
VIN+
1kΩ
0.5Vpk-pk
1Vpk-pk
ZIN = 2kΩ
Op amp wired as differential amplifier
1kΩ
0V

Fig.32. There is an interesting effect related to the input impedance of the differential amplifier; both input signals are equal and anti-phase. The 1kΩ resistor on the inverting input is reduced to an effective value of 667Ω or 2/3 of its value. This assumes all differeerntial amplifer resistors are equal.

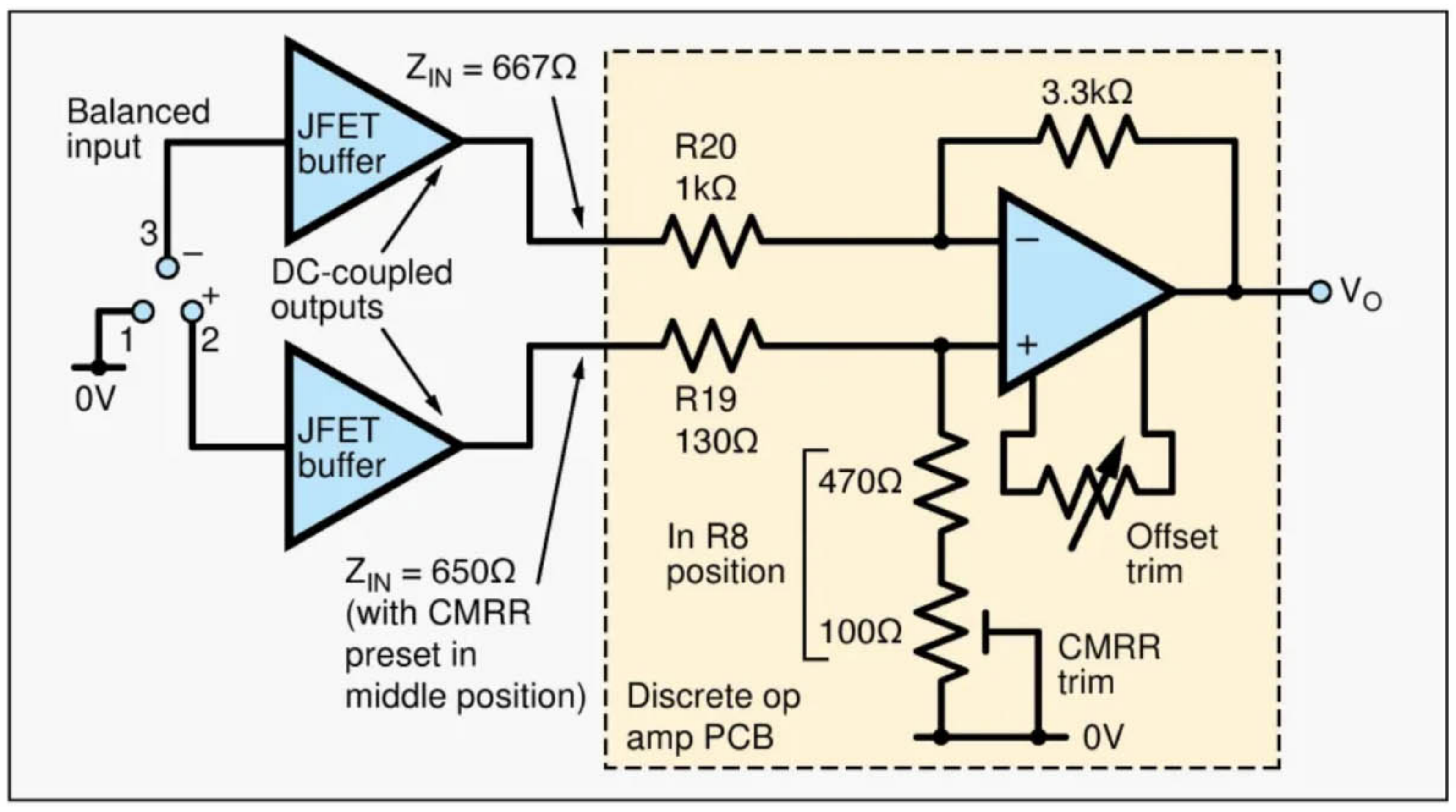

Balanced input
ZIN = 667Ω
JFET buffer
3
1
2
0V
DC-coupled outputs
JFET buffer
ZIN = 650Ω (with CMRR preset in middle position)
3.3kΩ
R20
1kΩ
R19
130Ω
470Ω
In R8 position
100Ω
VO
Offset trim
CMRR trim
0V
Discrete op amp PCB

Fig.33. In comparison with the circuit in Fig.32, the actual resistor values used take into account the resistance-reduction effect mentioned in the text. There is also a gain of 3.2, and the input impedances have been equalised. Note the preset for CMRR adjustment and the resistor values are low for noise minimisation.

DC coupling

The buffers can be DC coupled to the op amp inputs if desired since their +0.81V offset can be easily be rejected by the differential op amp. The offset adjust preset PR1 will need to be tweaked to bring the output of the whole system to zero. To do this, omit the bipolar DC blocking capacitor C8 from both buffer boards. By omitting these coupling capacitors, the CMRR will be maintained at low frequencies since electrolytics have poor tolerances, resulting in possible mismatching. However, there is the safety issue of a hard DC offset being transmitted to the output if there is a wiring error or fault in one of the buffers. This could then damage what is connected to the output. Of course, the output could be AC coupled, or have a DC detection circuit and muting relay put in. The final circuit is shown in Fig.33.

The assembly of the three PCBs is shown in Fig.34. They will go in a nice box one day to make a top-notch balanced headphone amp.

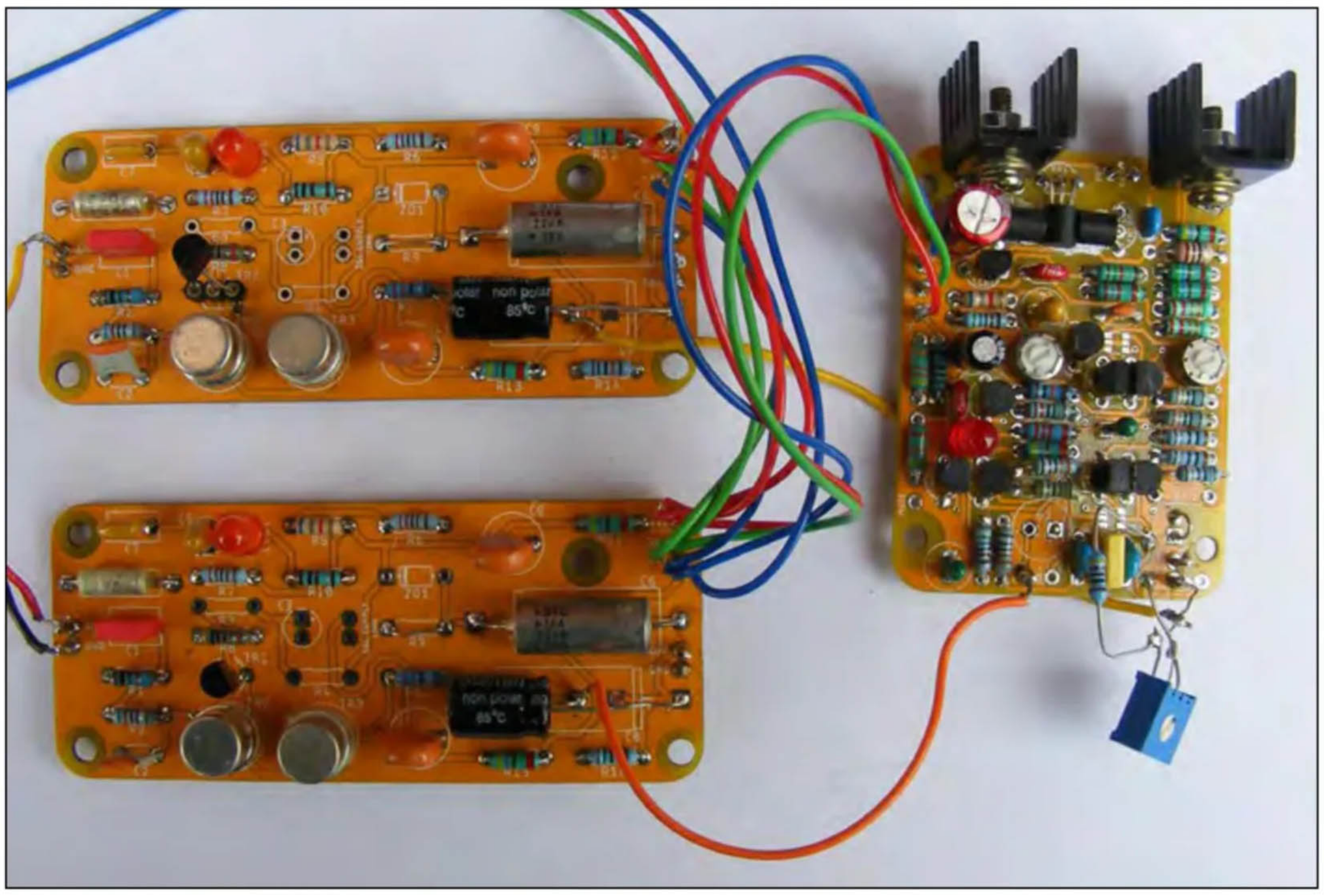


Fig.34. Connecting two buffer boards with a differential op amp to make an instrumentation or headphone balanced input amplifier. I hope to tidy this up and build it into a decent enclosure.

Next month

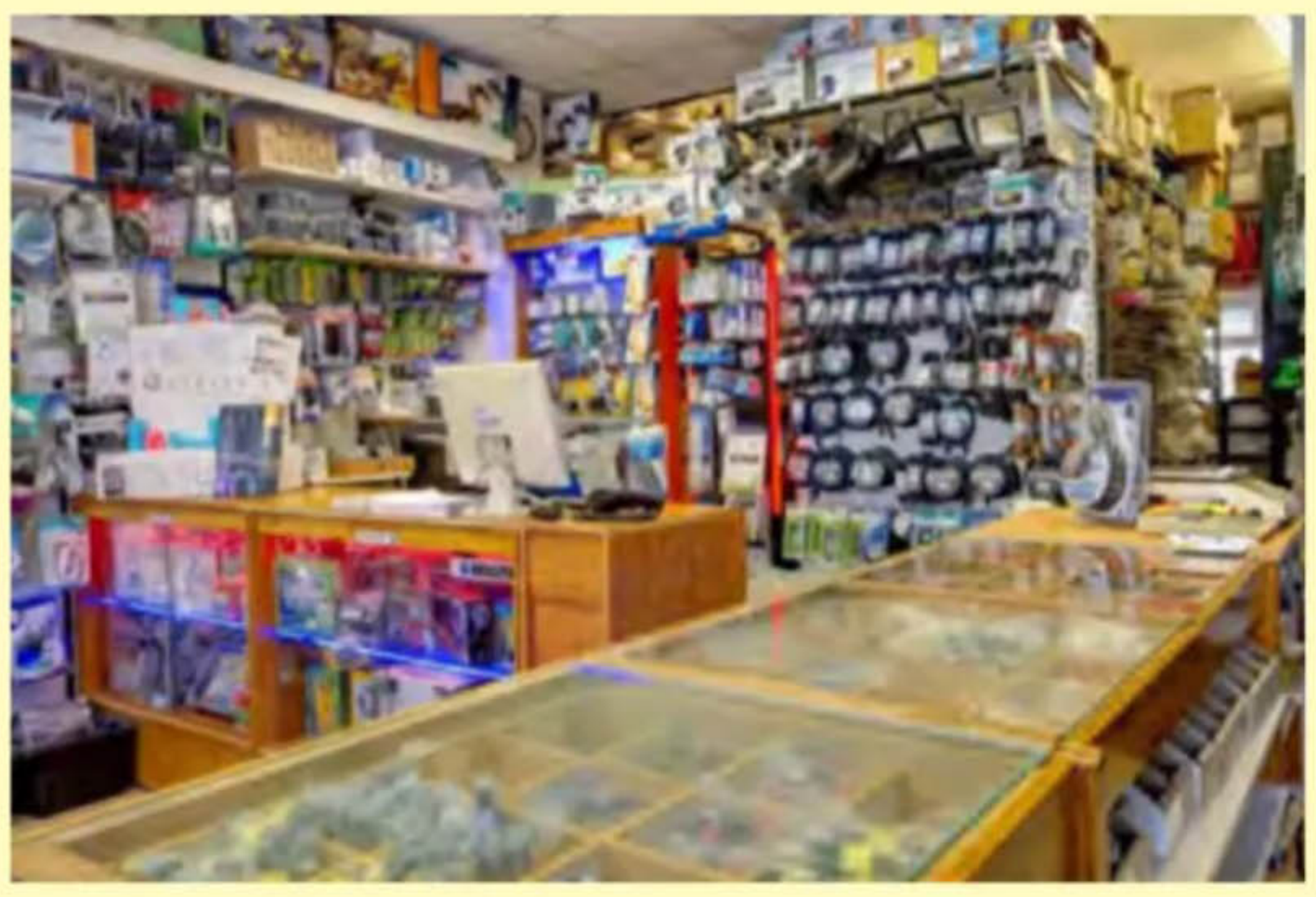
We will conclude our discrete buffer journey next month by looking at an op amp buffer circuit and an interesting design called the 'diamond buffer'.

ESR Electronic Components Ltd

All of our stock is RoHS compliant and CE approved. Visit our well stocked shop for all of your requirements or order on-line.
We can help and advise with your enquiry, from design to construction.

3D Printing • Cable • CCTV • Connectors • Components • Enclosures • Fans • Fuses • Hardware • Lamps • LED's • Leads • Loudspeakers • Panel Meters • PCB Production • Power Supplies • Relays • Resistors • Semiconductors • Soldering Irons • Switches • Test Equipment • Transformers and so much more…

Monday to Friday 08:30 - 17.00, Saturday 08:30 - 15:30



Station Road
Cullercoats
North Shields
Tyne & Wear
NE30 4PQ

Tel: 0191 2514363 sales@esr.co.uk www.esr.co.uk

BACK ISSUES

Practical Electronics

BACK ISSUES – ONLY £5.95

We can supply back issues of PE/EPE by post.

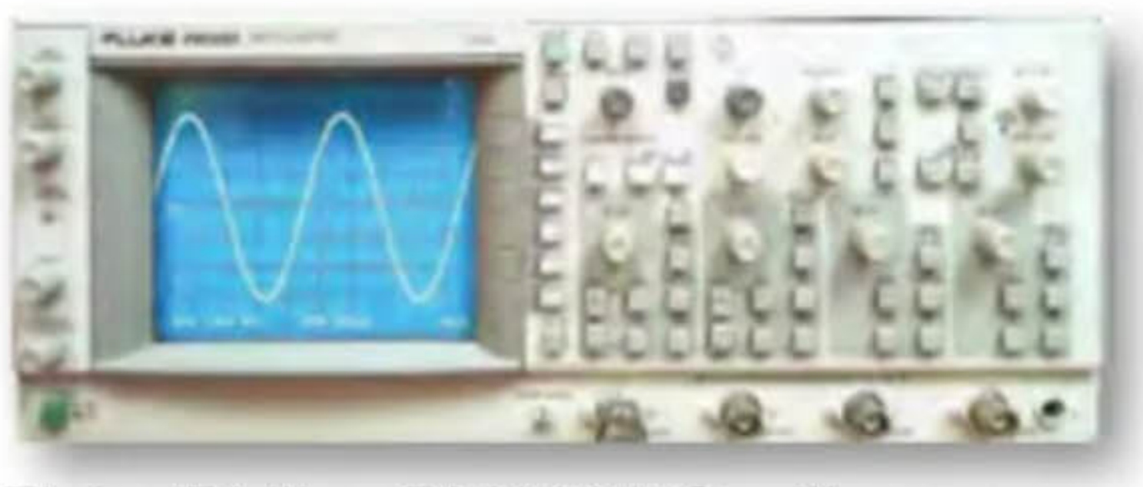
We stock magazines back to 2006, except for the following:

2006 Jan, Feb, Mar, Apr, May, Jul
2007 Jun, Jul, Aug
2008 Aug, Nov, Dec
2009 Jan, Mar, Apr
2010 May, Jun, Jul, Aug, Oct, Nov
2011 Jan
2014 Jan
2018 Jan, Nov, Dec
2019 Jan, Feb, Apr, May, Jun

Issues from Jan 1999 are available on CD-ROM / DVD-ROM

If we do not have a a paper version of a particular issue, then a PDF can be supplied – your email address must be included on your order.

Please make sure all components are still available before commencing any project from a back-dated issue.



Fluke/Philips PM3092 Oscilloscope
2+2 Channel 200MHz Delay TB,
Autoset etc – £250

STEWART OF READING

17A King Street, Mortimer, near Reading, RG7 3RS
Telephone: **0118 933 1111** Fax: **0118 933 2375**
USED ELECTRONIC TEST EQUIPMENT
Check website **www.stewart-of-reading.co.uk**
(ALL PRICES PLUS CARRIAGE & VAT)
Please check availability before ordering or calling in

HP 54600B Oscilloscope
Analogue/Digital Dual Trace 100MHz
Only £75, with accessories £125

LAMBDA GENESYS	PSU GEN100-15 100V 15A Boxed As New	£400
LAMBDA GENESYS	PSU GEN50-30 50V 30A	£400
IFR 2025	Signal Generator 9kHz – 2.51GHz Opt 04/11	£900
IFR 2948B	Communication Service Monitor Opts 03/25 Avionics	POA
IFR 6843	Microwave Systems Analyser 10MHz – 20GHz	POA
R&S APN62	Syn Function Generator 1Hz – 260kHz	£295
Agilent 8712ET	RF Network Analyser 300kHz – 1300MHz	POA
HP8903A/B	Audio Analyser	£750 – £950
HP8757D	Scaler Network Analyser	POA
HP3325A	Synthesised Function Generator	£195
HP3561A	Dynamic Signal Analyser	£650
HP6032A	PSU 0-60V 0-50A 1000W	£750
HP6622A	PSU 0-20V 4A Twice or 0-50V 2A Twice	£350
HP6624A	PSU 4 Outputs	£400
HP6632B	PSU 0-20V 0-5A	£195
HP6644A	PSU 0-60V 3.5A	£400
HP6654A	PSU 0-60V 0-9A	£500
HP8341A	Synthesised Sweep Generator 10MHz – 20GHz	£2,000
HP83630A	Synthesised Sweeper 10MHz – 26.5 GHz	POA
HP83624A	Synthesised Sweeper 2 – 20GHz	POA
HP8484A	Power Sensor 0.01-18GHz 3nW-10µW	£75
HP8560E	Spectrum Analyser Synthesised 30Hz – 2.9GHz	£1,750
HP8563A	Spectrum Analyser Synthesised 9kHz – 22GHz	£2,250
HP8566B	Spectrum Analsyer 100Hz – 22GHz	£1,200
HP8662A	RF Generator 10kHz – 1280MHz	£750
Marconi 2022E	Synthesised AM/FM Signal Generator 10kHz – 1.01GHz	£325
Marconi 2024	Synthesised Signal Generator 9kHz – 2.4GHz	£800
Marconi 2030	Synthesised Signal Generator 10kHz – 1.35GHz	£750
Marconi 2023A	Signal Generator 9kHz – 1.2GHz	£700
Marconi 2305	Modulation Meter	£250
Marconi 2440	Counter 20GHz	£295
Marconi 2945/A/B	Communications Test Set Various Options	POA
Marconi 2955	Radio Communications Test Set	£595
Marconi 2955A	Radio Communications Test Set	£725
Marconi 2955B	Radio Communications Test Set	£800
Marconi 6200	Microwave Test Set	£1,500
Marconi 6200A	Microwave Test Set 10MHz – 20GHz	£1,950
Marconi 6200B	Microwave Test Set	£2,300
Marconi 6960B	Power Meter with 6910 sensor	£295
Tektronix TDS3052B	Oscilloscope 500MHz 2.5GS/s	£1,250
Tektronix TDS3032	Oscilloscope 300MHz 2.5GS/s	£995
Tektronix TDS3012	Oscilloscope 2 Channel 100MHz 1.25GS/s	£450
Tektronix 2430A	Oscilloscope Dual Trace 150MHz 100MS/s	£350
Tektronix 2465B	Oscilloscope 4 Channel 400MHz	£600
Farnell AP60/50	PSU 0-60V 0-50A 1kW Switch Mode	£300
Farnell XA35/2T	PSU 0-35V 0-2A Twice Digital	£75
Farnell AP100-90	Power Supply 100V 90A	£900
Farnell LF1	Sine/Sq Oscillator 10Hz – 1MHz	£45
Racal 1991	Counter/Timer 160MHz 9 Digit	£150
Racal 2101	Counter 20GHz LED	£295
Racal 9300	True RMS Millivoltmeter 5Hz – 20MHz etc	£45
Racal 9300B	As 9300	£75
Solartron 7150/PLUS	6½ Digit DMM True RMS IEEE	£65/£75
Solatron 1253	Gain Phase Analyser 1mHz – 20kHz	£600
Solartron SI 1255	HF Frequency Response Analyser	POA
Tasakago TM035-2	PSU 0-35V 0-2A 2 Meters	£30
Thurlby PL320QMD	PSU 0-30V 0-2A Twice	£160 – £200
Thurlby TG210	Function Generator 0.002-2MHz TTL etc Kenwood Badged	£65

HP/Agilent HP 34401A Digital
Multimeter 6½ Digit £325 – £375

HP33120A	Function Generator 100 microHz – 15MHz	£350
HP53131A	Universal Counter 3GHz Boxed unused	£600
HP53131A	Universal Counter 225MHz	£350
Audio Precision	SYS2712 Audio Analyser – in original box	POA
Datron 4708	Autocal Multifunction Standard	POA
Druck DPI 515	Pressure Calibrator/Controller	£400
Datron 1081	Autocal Standards Multimeter	POA
ENI 325LA	RF Power Amplifier 250kHz – 150MHz 25W 50dB	POA
Keithley 228	Voltage/Current Source	POA
Time 9818	DC Current & Voltage Calibrator	POA

Marconi 2955B Radio
Communications Test Set – £800

AOShop

The home for specialist audio, analogue and historic components – provided by Jake Rothman, *PE's Audio Out* columnist. The AOShop is your best bet for classic analogue 'NOS' (new old stock) components, including all parts for *Audio Out* projects and designs.

Ge semiconductors

Small-signal PNP transistors

NKT214F, OC57, OC59, 2N1377, 2N525	£1.00

Low-noise PNP transistors

GET106	£2.50

Small power PNP transistors

AC153, AC153K, AC188, AC188K	£1.50

Small power NPN transistors

AC176, AC176K, AC187K, AC141K	£2.00

High-voltage PNP transistors

OC77, CV7001	£1.50

TO3 PNP power transistors

OC22, CV7054 (OC23), OC25, OC35, OC36, AD143, AD149, AD161, AD162	£2.00
AD140	£3.50
AD149, AD161/2 matched pair	£5.00

RF PNP transistors

OC41, OC42, 2SA12, 2SA53,AF124, AF178, GET872A	£1.50

NPN transistors

OC139, OC140, ASY73	£2.00

Diodes

CV7049 (OA10), CG92 (OA91)	£0.50

Si semiconductors

Diodes

ZC5800 RF Schottky	£0.20

Low-noise silicon transistors

BFW16A, 2SC3071, 2SC3068, 2SA1016K, 2SC2362K, 2SA970BL	£1.50
2SC2204, 2SD655, BC550C	£0.50
ZTX651	£0.30
ZTX751	£0.50

RF transistors (suitable for Theremin)

BF199	£0.50

Audio power MOSFETs

Exicon 10N20, 10P20	£6.50
Hitachi 2SJ99, 2SK343	£3.50
Hitachi 2SJ56, 2SK176	£8.50

JFETs

BFW11, BFW10, TIS73L, J177, J113, U1994, U1898, 2SJ176, J201	£1.00
J175, J176, J112, J111, 2N3820, 2N5467, BF244, 2N5460, J230	£0.60

Small power output/driver transistors

2SB649A, D669A, 2SA1208, 2SC2910, MJE253G, MJE243G, 2SA1725, 2SC4511	£1.20
BD139,BD140, BD135, BD156, BD435, BD436, MPSA63, BCV46	£0.50

MOSFETs

ZVP2106A	£0.30
Dual-gate 3SK45, BFS28	£1.50

Metal-cased transistors

BC143	£0.35
2N1711	£0.50
BCY71	£0.30
BC109C	£0.60

Amplifiers

LM318 high-speed op amp	£0.35
μA709 metal-cased op amp	£2.00
5534H metal-cased op amp	£3.00
LM384, TDA2030A, TDA2050V	£1.50
TAA435 (Mullard power amp driver)	£1.20

Synthesiser ICs

That/dbx 2180 VCA/VCF	£6.00
CA3080 VCA/VCF (vocoder)	£2.80
CA3280 VCA/VCF	£4.00
LM13600/ LM13700 VCA/VCF	£2.00
CA3086 transistor array	£1.00
PT2399 echo/delay	£1.50

Dual transistors

2N2639, 2N2223, 2N2910 (NPN)	£4.00
E401 (JFET Moog)	£4.00
2SK2145-Y dual JFET	£0.80
2N5564 JFET	£8.00
HN3C51F, HN3A51F	£1.00
DMMT3904/6, HN1A01F, HN1C01F	£0.50

Loudspeakers

PE Mini-Monitor

Volt PE165 6.5-inch woofer (each)	£85
Morel MDT29 tweeter (each)	£25
Kit *pair* of PE165/MDT29 plus Volt crossover parts and PCBs	£299
Monacor DT-28N tweeter (each)	£35
Vifa 19mm BC20SC15-04 tweeter (each)	£15

Volt crossover inductors

1.2mH, 1.5mH, 2mH, 2.7mH, 0.5mH (tapped at 0.3mH) (each)	£5.85

Fully assembled and tested high-quality speaker prototypes – ask for details

LS3/5As and other similar speaker systems (pair)	£200-£350

Fully tested reclaimed speakers

Vifa BC14 5-inch woofer (each)	£10
Vifa TC26 1-inch tweeter (each)	£10

Low-price speakers

Philips 4-inch 4070	£2.00
EMI 10x6-inch, 30Ω Alnico	£7
64mm 64Ω neodymium	£1.20
5x3-inch elliptical 50Ω or 80Ω Alnico	£3.50
1.65x2.75-inch 8Ω	£1.50

Capacitors

Note '10/63' denotes '10μF 63V'.

Polyester

3.3/100, 4.7/250, 4.7/63	£1.00
5.6/63, 8.2/63, 10/63	£2.00
Mullard 'Mustard' C296 0.22/400	£2.00

Polycarbonate

Axial 2.2/63 1%, 4.7/160, 6.8/63	£1.00

Radial

6.8/160V, 10/63	£2.00
Reclaimed 22/63	£2.00

Polystyrene

Philips 1% 4.7nF/160, 6.2nF/500, 12nF/63, 22nF/63, 110nF/63, 24nF, 2nF	£1.00
RIFA 1% 100nF/100, Suflex 90.9nF 0.5%	£2.00
Suflex 2.5% 10nF/63 (rad. or ax.)	£0.50

Radiation resistant

Siemens cellulose acetate MKL 2.2/25	£0.80

Electrolytic – Mullard blue 017 series

10/25, 22/25, 100/10	£0.50
150/40, 470/40, 1000/40	£1.00

Tantalum – axial metal cased

22/50, 47/35, 68/25, 100/20, 120/10, 150/16, 220/10, 330/6	£2.00
22/35, 33/35, 47/20, 68/15, 100/10, 150/6	£1.25
4.7/50, 6.8/35, 10/25, 10/35, 22/15	£1.00

Axial moulded-case tantalum

Kemet axial 6.8/10	£0.30
Kemet radial 33/10	£0.40
STC radial 100/20	£1.50

Tantalum bead

22/50	£1.00
470/3	£2.00
680/6.3	£3.00

Wet tantalum

220/25 axial	£3.20
Castanet button 140/30, 470/3	£3.20
Hughes 540/10	£3.20

Bipolar

Hermetic bipolar tantalum 16/35	£3.20
Elcap axial 10/50	£0.50
Generic radial 100/16, 470/35, 100/ 63, 22/35, 4.7/35, 220/16	£0.50

Philips solid-aluminium (axial) 121/123

47/16	£1.00
330/6.3	£2.00
100/35	£4.00

Philips Pearl 122 series (radial)

10/16	£0.30

Silvered mica (radial)

1nF/500 1%	£1.00

Trimmer capacitors

Vishay plastic-film 4-40pF	£0.80
Vishay plastic-film 5.5-45pF	£1.00
Vishay plastic-film 5-80pF	£1.20
Vishay plastic-film 10-250pF	£2.00
Mica 1-12pF, 2-40pF	£1.00

Audio transformers and inductors

Eagle transformers

LT44, LT722 driver, LT700, LT723 500Ω output	£2.50
LT30 500mW output	£3.50
5:1 interstage	£1.50

Repanco

T/T3 splitter transformer	£4.00
CH2 5mH RFC	£2.00

Balanced output transformer

Vigortronix 600Ω VTX-101-007	£10
Vigortronix 600Ω VTX-101-3001	£10
Vigortronix 600Ω VTX-101-3002	£15
Gardners 150Ω	£10
Reclaimed BBC LL74/MPC nickel core 600Ω	£12

Reclaimed mic input transformer	£15

Inductors

82μH, 4.7mH, 100μH, 270μH, 10μH, 14μH (low Z)	£0.50
7-inch ferrite rod with MW and LW windings	£3.00

Special resistors

Bourns wire-wound trimmer 5kΩ 3059 JM panel-mount	£2.00
Thermistor RA53, R13	£4.00
A13	£2.00
Thermistor CZ1, CZ6	£1.50
Holco H2 2.2MΩ 1W, 1%	£1.00
Welwyn 1GΩ 2W	£1.00
5k Bourns 3321H cermet trimmer	£0.50

Potentiometers

Bourns 81

25kΩ lin cermet	£2.00
25kΩ lin, 5kΩ lin conductive plastic	£3.00

Bourns 91

10k dual-gang lin or log	£10.00

Plessey moulded-track

5kΩ log with switch	£3.00
50kΩ A/log	£3.00
Mil 250kΩ lin dual	£5.00

Alpha

16mm 4.7kΩ A/log	£0.80
220kΩ A/log	£0.80
10kΩ lin centre-detent dual-gang	£1.50

Allen Bradley J series/Honeywell

10kΩ lin 1 million cycles	£5.00
Blore Edwards AB 45 dual 5kΩ A/log with switch	£3.50
Alps RK9 dual-gang 5kΩ RD law	£4.00
Alps 50k log tapped motorised stereo	£5.00
BI P260 500kΩ log conductive plastic 1 million cycles	£2.00
Colvern wire-wound 100kΩ or 50kΩ dual-gang 3W	£5.00
Mil Spec hermetic 10Ω	£8.00

Miscellaneous

Theremin Clearance Sale!

Elysian Theremin MIDI box	£300

PCBs

Pocket Theremin (*EPE*, 1996)	£2.00
Elysian Theremin (*EPE*, 1996)	£6
Synth VCF, VCO (*EPE*, 2017)	£3.00
48V PSU (*EPE*, 2019)	£3.00
SMT dual transistor adapter	£0.40

Contact

Jake Rothman
The Old Rectory, Arlais Road,
Llandrindod Wells, Powys
LD1 5HE (visit by appointment)

+44 (0)1597 829102

jrothman1962@gmail.com

Minimum order £5.00 inc post
Quantity discounts negotiable

Payment

PayPal, cards (via phone), bank transfer, cheques (payable to 'J Rothman', UK pounds only)
No VAT payable

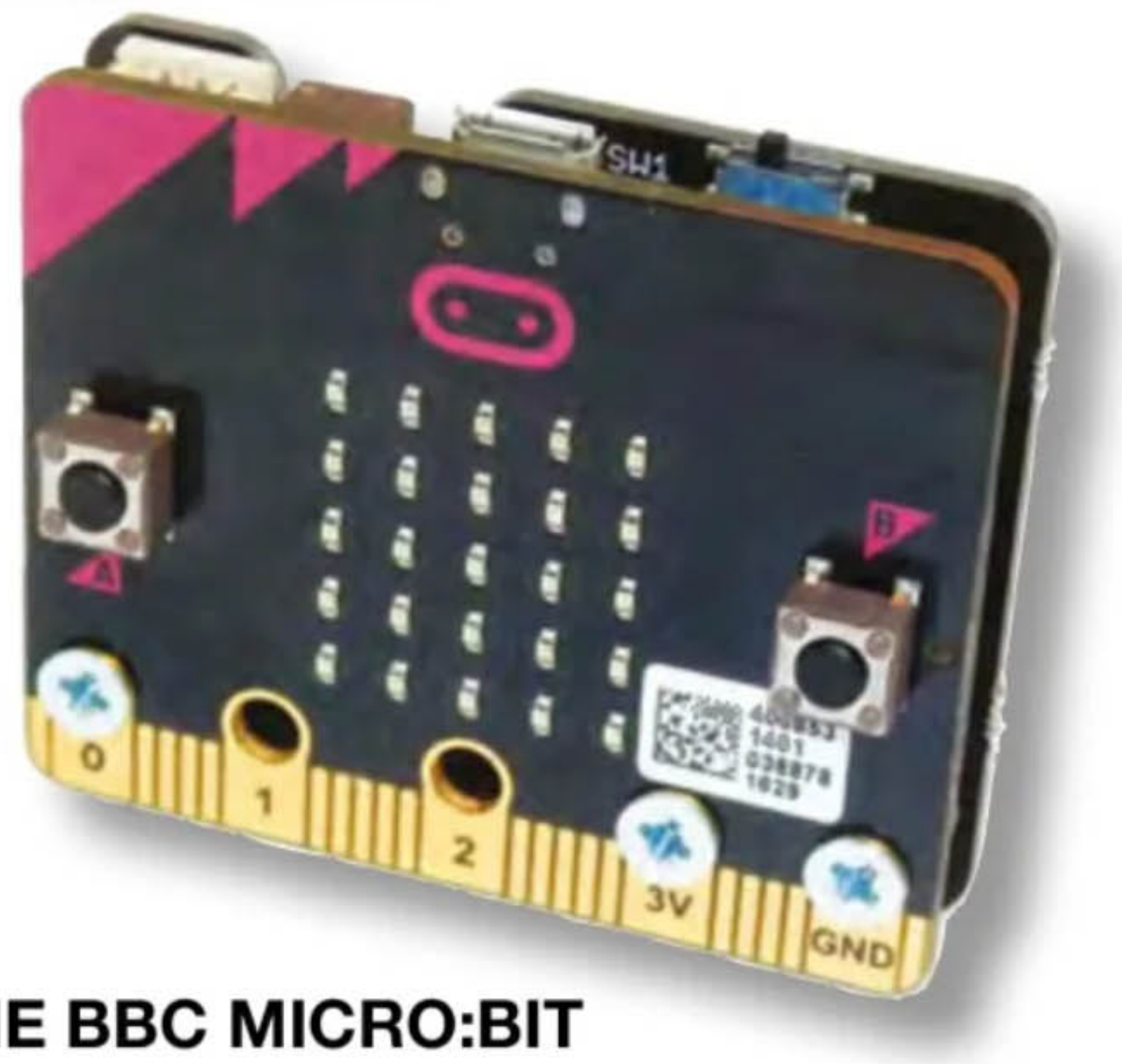
Postage

Small Jiffy bag £2.99
Small package £4.99
Big boxes and overseas at cost – ask for a quote

DIRECT BOOK SERVICE

Teach-In 2017

Introducing the BBC micro:bit



IN FULL COLOUR
For Beginners of Any Age
Python Coding on the BBC micro:bit
Jim Gatenby

PYTHON CODING ON THE BBC MICRO:BIT

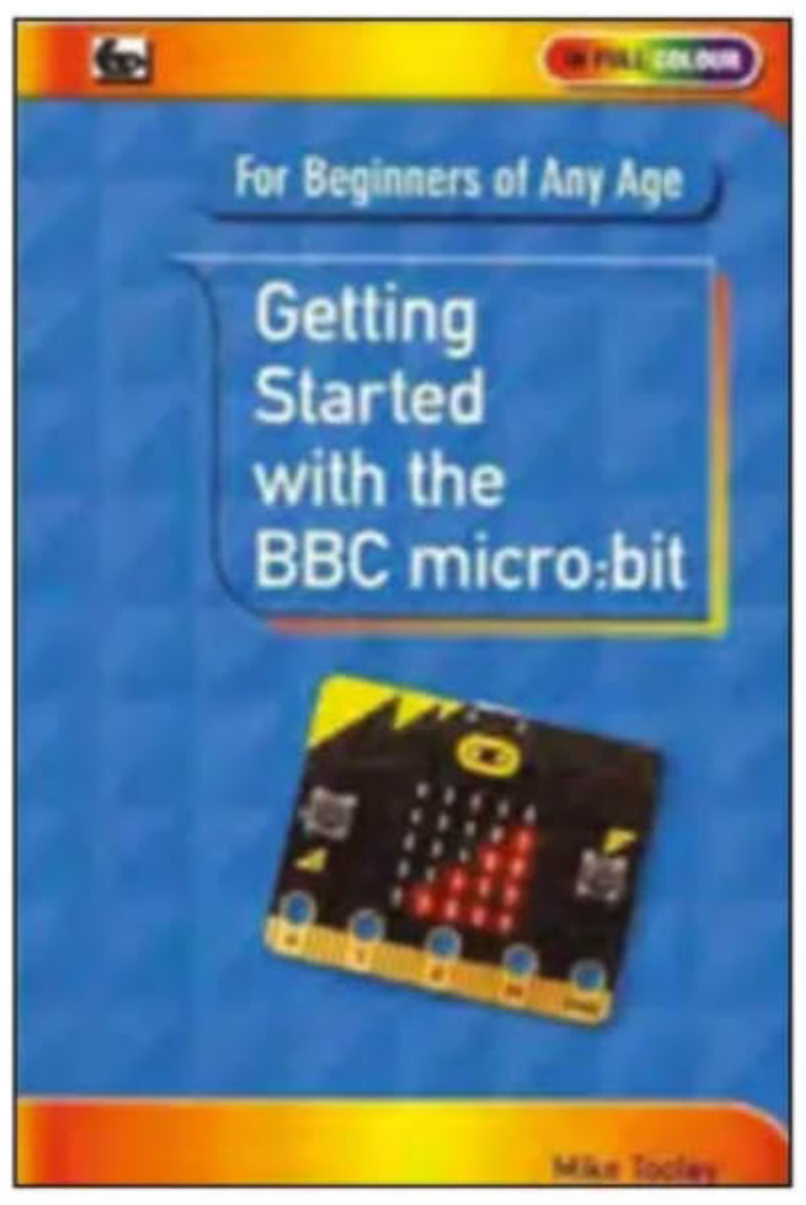
Jim Gatenby

Python is the leading programming language, easy to learn and widely used by professional programmers. This book uses MicroPython, a version of Python adapted for the BBC Micro:bit.

Among the many topics covered are: main features of the BBC micro:bit including a simulation in a web browser screen; various levels of programming languages; Mu Editor for writing, saving and retrieving programs, with sample programs and practice exercises; REPL, an interactive program for quickly testing lines of code; scrolling messages, creating and animating images on the micro:bit's LEDs; playing and creating music, sounds and synthesized speech; using the on-board accelerometer to detect movement of the micro:bit on three axes; glossary of computing terms.

This book is written using plain English, avoids technical jargon wherever possible and covers many of the coding instructions and methods which are common to most programming languages. It should be helpful to beginners of any age, whether planning a career in computing or writing code as an enjoyable hobby.

118 Pages **Order code PYTH MBIT** **£7.99**



GETTING STARTED WITH THE BBC MICRO:BIT

Mike Tooley

Not just an educational resource for teaching youngsters coding, the BBC micro:bit is a tiny low cost, low-profile ARM-based single-board computer. The board measures 43mm × 52mm but despite its diminutive footprint it has all the features of a fully fledged microcontroller together with a simple LED matrix display, two buttons, an accelerometer and a magnetometer.

Mike Tooley's book will show you how the micro:bit can be used in a wide range of applications from simple domestic gadgets to more complex control systems such as those used for lighting, central heating and security applications. Using Microsoft Code Blocks, the book provides a progressive introduction to coding as well as interfacing with sensors and transducers.

Each chapter concludes with a simple practical project that puts into practice what the reader has learned. The featured projects include an electronic direction finder, frost alarm, reaction tester, battery checker, thermostatic controller and a passive infrared (PIR) security alarm.

No previous coding experience is assumed, making this book ideal for complete beginners as well as those with some previous knowledge. Self-test questions are provided at the end of each chapter, together with answers at the end of the book. So whatever your starting point, this book will take you further along the road to developing and coding your own real-world applications.

108 Pages **Order code BBC MBIT** **£7.99**

The books listed here have been selected by the *Practical Electronics* editorial staff as being of special interest to everyone involved in electronics and computing. They are supplied by mail order direct to your door.

FOR A FULL DESCRIPTION OF THESE BOOKS AND CD-ROMS SEE THE SHOP ON OUR WEBSITE

www.electronpublishing.com

All prices include UK postage

MICROPROCESSORS

INTERFACING PIC MICROCONTROLLERS – 2nd Ed
Martin Bates
298 pages **Order code NE48** **£34.99**

PROGRAMMING 16-BIT PIC MICROCONTROLLERS IN C – LEARNING TO FLY THE PIC24
Lucio Di Jasio (Application Segments Manager, Microchip, USA)
496 pages + CD-ROM **Order code NE45** **£38.00**

INTRODUCTION TO MICROPROCESSORS AND MICROCONTROLLERS – 2nd Ed
John Crisp
222 pages **Order code NE31** **£29.99**

THE PIC MICROCONTROLLER YOUR PERSONAL INTRODUCTORY COURSE – 3rd Ed
John Morton
270 pages **Order code NE36** **£25.00**

PIC IN PRACTICE – 2nd Ed
David W. Smith
308 pages **Order code NE39** **£24.99**

MICROCONTROLLER COOKBOOK
Mike James
240 pages **Order code NE26** **£36.99**

THEORY AND REFERENCE

PRACTICAL ELECTRONICS HANDBOOK – 6th Ed
Ian Sinclair
440 pages **Order code NE21** **£33.99**

STARTING ELECTRONICS – 4th Ed
Keith Brindley
296 pages **Order code ELSEV100** **£18.99**

ELECTRONIC CIRCUITS – FUNDAMENTALS & APPLICATIONS – Updated version
Mike Tooley
400 pages **Order code TF43** **£32.99**

FUNDAMENTAL ELECTRICAL AND ELECTRONIC PRINCIPLES – 3rd Ed
C.R. Robertson
368 pages **Order code TF47** **£21.99**

A BEGINNER'S GUIDE TO TTL DIGITAL ICs
Robert Penfold
142 pages **OUT OF PRINT BP332** **£5.45**

UNDERSTANDING ELECTRONIC CONTROL SYSTEMS
Owen Bishop
228 pages **Order code NE35** **£36.99**

BOOK ORDERING DETAILS

All prices include UK postage.

For postage, add £3 per book to Europe, £4 for rest of the world per book.

CD-ROM prices include VAT and/or postage to anywhere in the world.

Send a cheque, (£ sterling only) made payable to: **Practical Electronics** or credit card details (Visa or Mastercard) to:

Electron Publishing Limited,
113 Lynwood Drive, Wimborne, Dorset BH21 1UU

Books are normally sent within seven days of receipt of order.

Please check price (see latest issue of Practical Electronics or website) before ordering from old lists.

For a full description of these books please see the shop on our website.
Tel: 01202 880299 – Email: shop@electronpublishing.com

Order from our online shop at: www.electronpublishing.com

Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
APRIL 2024		
Basic RF Signal Generator	CSE221001	8.99
Basic RF Signal Generator (black front panel)	CSE220902B	9.99
Silicon Chirp Cricket	08101231	7.99
Test Bench 'Swiss Army Knife'	04110221	9.99
MARCH 2024		
Digital Volume Control Potentiometer SMD version	01101231	8.95
Digital Volume Control Potentiometer through-hole version	01101232	9.95
Advanced SMD Test Tweezers body	04106221	6.95
Advanced SMD Test Tweezers legs (pair)	04106212	7.95
FEBRUARY 2024		
Active Mains Soft Starter	10110221	10.95
JANUARY 2024		
Q Meter	CSE220701	8.95
Q Meter (black solder mask)	CSE220704	7.95
Raspberry Pi Pico W BackPack	07101221	8.95
DECEMBER 2023		
Digital Boost Regulator	24110224	8.95
Dual-Channel Power Supply for Breadboards	04112221	11.95
Display Adaptor for the Breadboard PSU	04112222	9.95
NOVEMBER 2023		
LC Meter Mk3	CSE220503C	9.95
DC Supply Filter for vehicles	08108221	8.95
Discrete Audio Op Amp PCB	AO1-JUL23	9.95
Discrete Audio Op Amp PCB + essential components	AO2-JUL23	17.95
OCTOBER 2023		
Buck/Boost Charger Adaptor	14108221	9.95
PIC Breakout Board for SOIC parts	24110225	5.95
PIC Breakout Board for DIP parts	24110222	5.95
AVR64DD32 Breakout board	24110223	5.95
Automatic Train Controller	09109221	6.95
Chuff Sound module	09109222	6.95
SEPTEMBER 2023		
Mini LED Driver	16106221	7.95
New GPS-Synchronised Clock	19109221	9.95
Wide-Range Ohmmeter	04109221	14.95
AUGUST 2023		
110dB RF Attenuator	CSE211003	9.95
Universal Battery Charge Controller (2023 update)	14107192	9.95
Wide-Range OhmMeter	04109221	12.95
JULY 2023		
Multimeter Checker/Calibrator	04107221	10.95
MIDI Spectral Sound Synthesiser (full kit – see p.25)	N/A	N/A
JUNE 2023		
Arduino Programmable Load	04105221	8.95
Buck-Boost LED Driver	16103221	8.95
MAY 2023		
Precision AM-FM DDS Signal Generator	CSE211002	14.95
Improved SMD Test Tweezers programmed PIC	0410621PIC	12.95
AO 2x-dual-to-quad through-hole (pack of five)	AO1-MAY23	6.95
AO 2x-dual-to-quad SMD (pack of five)	AO2-MAY23	6.95
AO 2x-single-to-dual through-hole (pack of five)	AO3-MAY23	6.95
AO 2x-single-to-dual SMD (pack of five)	AO4-MAY23	6.95

PROJECT	CODE	PRICE
APRIL 2023		
500W Amplifier Module	see p.22, April 2023	
Clipping Indicator (per channel)	01112211	7.95
CD Welder Power Supply (one needed)	29103221	9.95
CD Welder Controller (one needed)	29103222	9.95
CD Welder Energy Storage module (several needed)	29103223	7.95
AO Universal Dual Op Amp Board	AO1-APR23	9.95
AO Stereo RIAA precision passives kit for dual op amp	AO2-APR23	8.95
MARCH 2023		
Pico BackPack	07101221	9.95
Semaphore Signal (controller)	09103221	7.95
Semaphore Signal (blade)	09103222	5.95
FEBRUARY 2023		
Tesla Coil driver board	26102221	9.95
Tesla Coil potentiometer board	26102222	5.95
Cooling Fan Controller & Loudspeaker Protector	01102221	8.95
Remote Gate Controller	11009121	12.95
JANUARY 2023		
Classic LED Metronome – 8-LED	23111211	7.95
Classic LED Metronome – 10-LED	23111212	8.95
Multi-Channel Speaker Protector – 6 channel	01101221	9.95
Multi-Channel Speaker Protector – 4 channel	01101222	7.95
Remote Control Range Extender – IR-to-UHF	15109212	5.95
Remote Control Range Extender – UHF-to-IR	15109211	7.95
AO Universal Single Op Amp Board	AO1-JAN23	7.95
DECEMBER 2022		
Hummingbird Amplifier	01111211	9.95
SMD Trainer PCB	29106211A	8.95
SMD Trainer PCB + parts	29106211B	13.95
NOVEMBER 2022		
USB Cable tester – main PCB	04108211	12.95
USB Cable tester – front panel	04108212	5.95
USB Cable tester – optional panel	SC5970	5.95
Model Railway Carriage Lights – PCB	09109211	6.95
AO transfmr PCB – standard VTX-A range	VTX-101-007	6.95
AO transfmr PCB – dual-outline VTX102-3001/101-3002	VTX-Dual	6.95
OCTOBER 2022		
SMD Test Tweezers – PCB and pair of tweezer arms	04106211-2	11.95
SMD Test Tweezers – programmed PIC12F1572-I/SN	0410621A	7.95
Tele-com	12110211	12.95
SEPTEMBER 2022		
Touchscreen Digital Preamp – main board	01103191	12.95
Touchscreen Digital Preamp – adaptor board pair	01103192	5.95
20A DC Motor Speed Controller	11006211	9.95
AUGUST 2022		
Multi-purpose Battery Manager – I/O Expander module	11104212	5.95
Multi-purpose Battery Manager – Switch Module	11104211	8.95
Simple MIDI Music Keyboard (for 8 switches)	23101213	6.95
Nano Pong	08105212	7.95
JULY 2022		
Silicon Labs AM/FM/SW Radio	CSE210301C	10.95
Level Crossing Controller	09108211	6.95
JUNE 2022		
Full-wave Motor Speed Controller	1010221	8.95
PIC Programming Helper for 8-pin PICs only	24106211	7.95
PIC Programming Helper for 8, 14 or 20-pin PICs	24106212	10.95
Advanced GPS Computer	05102211	9.95

PCBs for most recent *PE*/*EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.

Back issues of articles are available – see Back Issues page for details.

Double-sided | plated-through holes | solder mask

PROJECT	CODE	PRICE
MAY 2022		
Bus board PCB for Analogue Vocoder	AO1-MAY22	10.95
Complete set of 14 PCBs for Analogue Vocoder	AO2-MAY22	97.95
Programmed EEPROM for Digital FX Unit	FX1-MAY22	10.95
Programmed PIC for Digital FX Unit using potentiometer	FX2-MAY22	8.95
APRIL 2022		
64-key MIDI Matrix shield	23101211	8.95
64-key MIDI Matrix switch board	23101212	11.95
High-current Battery Balancer	14102211	10.95
Digital FX Unit – using potentiometer	01102211	9.95
Digital FX Unit – using BCD switch	01102212	9.95
Universal Audio PSU	AO1-APR22	11.95
MARCH 2022		
Mini Isolated Serial Link	24102211	£5.95
Busy Loo Indicator	16112201	£5.95
Analogue Vocoder – Band-pass filter board	AO1-MAR22	9.95
Analogue Vocoder – HP/LP filter board	AO2-MAR22	9.95
FEBRUARY 2022		
Arduino-based Power Supply	18106201	9.95
Battery Monitor Logger	11106201	10.95
Electronic Wind Chimes	23011201	10.95
Analogue Vocoder – Driver Amplifier	AO-FEB22	8.95
JANUARY 2022		
Vintage battery Radio Li-ion Power Supply	11111201	9.95
MiniHeart: A Miniature Heartbeat Simulator	01109201	8.95
DECEMBER 2021		
AM/FM/SW Digital Receiver	CSE200902A	13.95
Balanced Input and Attenuator for USB CODEC	01106202	11.95
NOVEMBER 2021		
Dual Battery Lifesaver	11111202	£6.95
OCTOBER 2021		
Mini Wi-Fi LCD BackPack	24106201	£8.95
SEPTEMBER 2021		
USB SuperCodec PCB	01106201	£14.95
Audio DDS Oscillator PCB	01110201	£5.95
Audio DDS Oscillator rotary encoder	01110201-ENC	6.95
Programming Adaptor Board for Audio DDS Oscillator	01110202	£5.95
High-power Ultrasonic Cleaner main PCB	04105201	£14.95
High-power Ultrasonic Cleaner front-panel PCB	04105202	
Night Keeper Lighthouse PCB	08110201	£5.95
AUGUST 2021		
Ol' Timer PCB	19104201	£11.95
Ol' Timer 8x8 RGB LED module using WS2812B	19104201-88	£8.95
Ol' Timer set of acrylic case pieces and spacer	19104201-ACR	£8.75
Ol' Timer DS3231 RTC IC wide SOIC-16	19104201-RTC	£5.95
Wideband Digital RF Power Meter	04106201	£9.75
Switchmode 78xx regulators ***(PACK of 5!)***	18105201	£7.95
Cool Beans SMAD display	CB-AUG21	£11.95
JULY 2021		
ATtiny816 Breakout / Dev Board with Capacitive Touch	24110181	£9.75
IR Remote Control Assistant (Jaycar version)	15005201	£8.95
IR Remote Control Assistant (Altronics version)	15005202	£8.95
PIC18F Development Board	PNM-JUL21	£12.95
Microphone Preamplifier	AO-JUL21	£11.95

PROJECT	CODE	PRICE
JUNE 2021		
Roadies' Test Signal Generator (surface-mount version)	01005201	£8.95
Roadies' Test Signal Generator (through-hole version)	01005202	£9.95
Touchscreen Wide-range RCL Box (Resistor module)	04104201	£18.95
Touchscreen Wide-range RCL Box (Ind/Cap module)	04104202	
KickStart Part 3 – Gyrator-based Audio Filter	KS3-2021	£7.95
MAY 2021		
7-Band Equaliser (Mono)	01104201	£8.95
7-Band Equaliser (Stereo)	01104202	£10.95
Car Altimeter	05105201	£7.95
APRIL 2021		
Reflow Oven – DSP Active Crossover (CPU)	01106193	£19.95
Reflow Oven – DSP Active Crossover (Front panel)	01106195	
Reflow Oven – DSP Active Crossover (LCD)	01106196	
Frequency Reference Signal Distributor	CSE200103	£8.95
MARCH 2021		
Nutube Guitar Effects Pedal	01102201	£12.95
Programmable Thermal Regulator (Peltier Interface)	21109181	£18.95
Programmable Thermal Regulator (Peltier Driver)	21109182	
Tunable HF Preamp	CSE190502	£8.95

For the many pre-2016 PCBs that we stock please see the PE website: www.electronpublishing.com

PE/EPE PCB SERVICE

Order Code	Project	Quantity	Price

Name ....

Address ....

....

Tel ....

Email ....

I enclose payment of £ .... (cheque/PO in **£ sterling only**)

payable to: **Practical Electronics**

Card No ....

Valid From .... Expiry Date ....

Card Security No ....

MasterCard VISA

You can also order PCBs by phone, email or via the shop on our website: www.electronpublishing.com

No need to cut your issue – a copy of this form is just as good!

All prices include VAT and UK p&p. Add £4 per *project* for post to Europe; £5 per *project* outside Europe.

Orders and payment should be sent to:

Practical Electronics, Electron Publishing Ltd
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU
Tel 01202 880299 Email: shop@electronpublishing.com
On-line Shop: www.epemag.com

Cheques should be made payable to 'Practical Electronics' **(Payment in £ sterling only)**.

NOTE: Most boards are in stock and sent within seven days of receipt of order, please allow up to 28 days delivery if we need to restock.

Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
FEBRUARY 2021		
4G Remote Monitoring	27111191	£9.95
JANUARY 2021		
Nutube Valve Preamplifier	01112191	£12.95
Arduino DCC Controller	09207181	£10.95
DECEMBER 2020		
Pseudo-Random Sequence Generator	16106191	£7.95
Clever Charger	14107191	£11.95
PE Theremin Amplifier	AO-1220-01	£8.95
NOVEMBER 2020		
LED Christmas Tree (1 off)	16107181-1	£6.95
LED Christmas Tree (4 off)	16107181-2	£14.95
LED Christmas Tree (12 off)	16107181-3	£24.95
LED Christmas Tree (20 off)	16107181-4	£34.95
USB/SPI Interface Board	16107182	£5.95
45V/8A Power Supply PCB *plus* acrylic spacer	18111181	£14.95
45V/8A Power Supply front panel five-way display bezel	18111181-BZ	£3.95
Five-way LCD Panel Meter/Display	18111182	£7.95
OCTOBER 2020		
Digital Audio Millivoltmeter	04108191	£9.95
Precision Signal Amplifier	04107191	£7.95
SEPTEMBER 2020		
PE Theremin PSU	AO-0920-01	£5.95
PE Theremin PSU transformer	AO-0920-02	£7.95
Micromite Explore-28	07108191	£6.95
Ultrabrite LED Driver	16109191	£6.95
AUGUST 2020		
Micromite LCD BackPack V3	07106191	£9.95
Steering Wheel Audio Button to Infrared Adaptor	05105191	£7.95
JULY 2020		
AM/FM/CW Scanning HF/VHF RF Signal Generator	04106191	£13.95
Speech Synthesiser with the Raspberry Pi Zero	01106191	£5.95
PE Mini-organ PCB	AO-0720-01	£14.95
PE Mini-organ selected parts	AO-0720-02	£8.95
High-current Solid-state 12V Battery Isolator – control	05106191	£6.95
High-current Solid-state 12V Battery Isolator FET (2oz)	05106192	£9.95
JUNE 2020		
Arduino breakout board – 3.5-inch LCD Display	24111181	£6.95
Six-input Audio Selector main board	01110191	£10.95
Six-input Audio Selector switch panel board	01110192	
MAY 2020		
Ultra-low-distortion Preamplifier Input Selector	01111112	£11.25
Ultra-low-distortion Preamplifier pushbutton Input Selector	01111113	
Universal Regulator	18103111	£7.95
433MHz Wireless Data Repeater	15004191	£8.50
Bridge-mode Adaptor for Amplifier	01105191	£7.95
iCEstick VGA Terminal	02103191	£5.95
Analogue noise with tilt control	AO-0520-01	£7.95
Audio Spectrum Analyser	PM-0520-01	£8.95
APRIL 2020		
Flip-dot Display black coil board	19111181	£14.95
Flip-dot Display black pixels	19111182	
Flip-dot Display black frame	19111183	
Flip-dot Display green driver board	19111184	

PROJECT	CODE	PRICE
MARCH 2020		
Diode Curve Plotter	04112181	£10.95
Steam Train Whistle / Diesel Horn Sound Generator	09106181	£8.50
Universal Passive Crossover (one off)	UPC0320	£12.50
FEBRUARY 2020		
Motion-Sensing 12V Power Switch	05102191	£5.95
USB Keyboard / Mouse Adaptor	24311181	£8.50
DSP Active Crossover (ADC)	01106191	£29.95
DSP Active Crossover (DAC) ×2	01106192	
DSP Active Crossover (CPU)	01106193	
DSP Active Crossover (Power/routing)	01106194	
DSP Active Crossover (Front panel)	01106195	
DSP Active Crossover (LCD)	01106196	
JANUARY 2020		
Isolated Serial Link	24107181	£8.50
DECEMBER 2019		
Extremely Sensitive Magnetometer	04101011	£16.75
Four-channel High-current DC Fan and Pump Controller	05108181	£8.75
Useless Box	08111181	£11.50
NOVEMBER 2019		
Tinnitus & Insomnia Killer (Jaycar case – see text)	01110181	£9.95
Tinnitus & Insomnia Killer (Altronics case – see text)	01110182	£9.95
OCTOBER 2019		
Programmable GPS-synced Frequency Reference	04107181	£11.50
Digital Command Control Programmer for Decoders	09107181	£9.95
Opto-isolated Mains Relay (main board)	10107181	£11.50
Opto-isolated Mains Relay (2 × terminal extension board)	10107182	
AUGUST 2019		
Brainwave Monitor	25108181	£12.90
Super Digital Sound Effects Module	01107181	£6.95
Watchdog Alarm	03107181	£8.00
PE Theremin (three boards: pitch, volume, VCA)	PETX0819	£19.50
PE Theremin component pack (see p.56, August 2019)	PETY0819	£15.00
JULY 2019		
Full-wave 10A Universal Motor Speed Controller	10102181	£12.90
Recurring Event Reminder	19107181	£8.00
Temperature Switch Mk2	05105181	£10.45
JUNE 2019		
Arduino-based LC Meter	04106181	£8.00
USB Flexitimer	19106181	£10.45
MAY 2019		
2× 12V Battery Balancer	14106181	£5.95
Deluxe Frequency Switch	05104181	£10.45
USB Port Protector	07105181	£5.95
APRIL 2019		
Heater Controller	10104181	£14.00
MARCH 2019		
10-LED Bargraph Main Board	04101181	£11.25
+Processing Board	04101182	£8.60
FEBRUARY 2019		
1.5kW Induction Motor Speed Controller	10105122	£24.95
NOVEMBER 2018		
Super-7 AM Radio Receiver	06111171	£15.95

CLASSIFIED ADVERTISING

Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value. The rate for semi-display space is £10 (+VAT) per centimetre high, with a minimum height of 2·5cm. All semi-display adverts have a width of 5.5cm. The prepaid rate for classified adverts is 40p (+VAT) per word (minimum 12 words).

Cheques are made payable to 'Practical Electronics'. **VAT must be added**. Advertisements with remittance should be sent to: Practical Electronics, 113 Lynwood Drive, Wimborne, Dorset, BH21 1UU.

Tel 07973518682 Email: pe@electronpublishing.com

For rates and further information on display and classified advertising please contact our Advertisement Manager, Matt Pulzer – see below.

Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.

We have been the leading monthly magazine in this market for the last twenty-seven years.

BOWOOD ELECTRONICS LTD
Suppliers of Electronic Components
www.bowood-electronics.co.uk
Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield, Derbyshire S40 2QR. Sales: 01246 200 222
Send large letter stamp for Catalogue

MISCELLANEOUS

VALVES AND ALLIED COMPONENTS?
For free stock list and/or advice, please contact me: geoffdavies337@gmail.com
Telephone: 01788 574774

PIC DEVELOPMENT KITS, DTMF kits and modules, CTCSS Encoder and Decoder/Display kits.
Visit www.cstech.co.uk

Electrical Industries Charity (EIC)
We help people working in the electrical, electronics and energy community as well as their family members and retirees.

We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.
www.electricalcharity.org

electrical industries charity

COMPONENTS AS JOB LOT
Large stock. Transistors, Diodes, Logic, Memory and Analogue parts. 1000s of passive items. Catalogued and sorted. 'New Old Stock' in racks.
Email: rojobarn@gmail.com

COAST ELECTRONICS
BREAKOUTS-COMPONENTS-CONTRACT DESIGN-3D PRINTER PARTS-MUSICAL-MICROCONTROLLERS
WWW.COASTELECTRONICS.CO.UK

Andrew Kenny – Qualified Patent Agent
EPO UKIPO USPTO
Circuits Electric Machinery Mechatronics
Web: www.akennypatentm.com
Email: Enquiries@akennypatentm.com
Tel: 0789 606 9725

ADVERTISING INDEX

CRICKLEWOOD ELECTRONICS	55
ESR ELECTRONIC COMPONENTS	65
FLOWCODE	15
HAMMOND ELECTRONICS Ltd	11
JPG ELECTRONICS	72
MICROCHIP	Cover (ii)
PEAK ELECTRONIC DESIGN	Cover (iv)
POLABS D.O.O.	22
QUASAR ELECTRONICS	3
SILICON CHIP	5
STEWART OF READING	65
TAG-CONNECT	55
TERRINGTON COMPONENTS	13

Advertisement offices

Matt Pulzer

Electron Publishing Ltd
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Tel 07973 518682

Email pe@electronpublishing.com

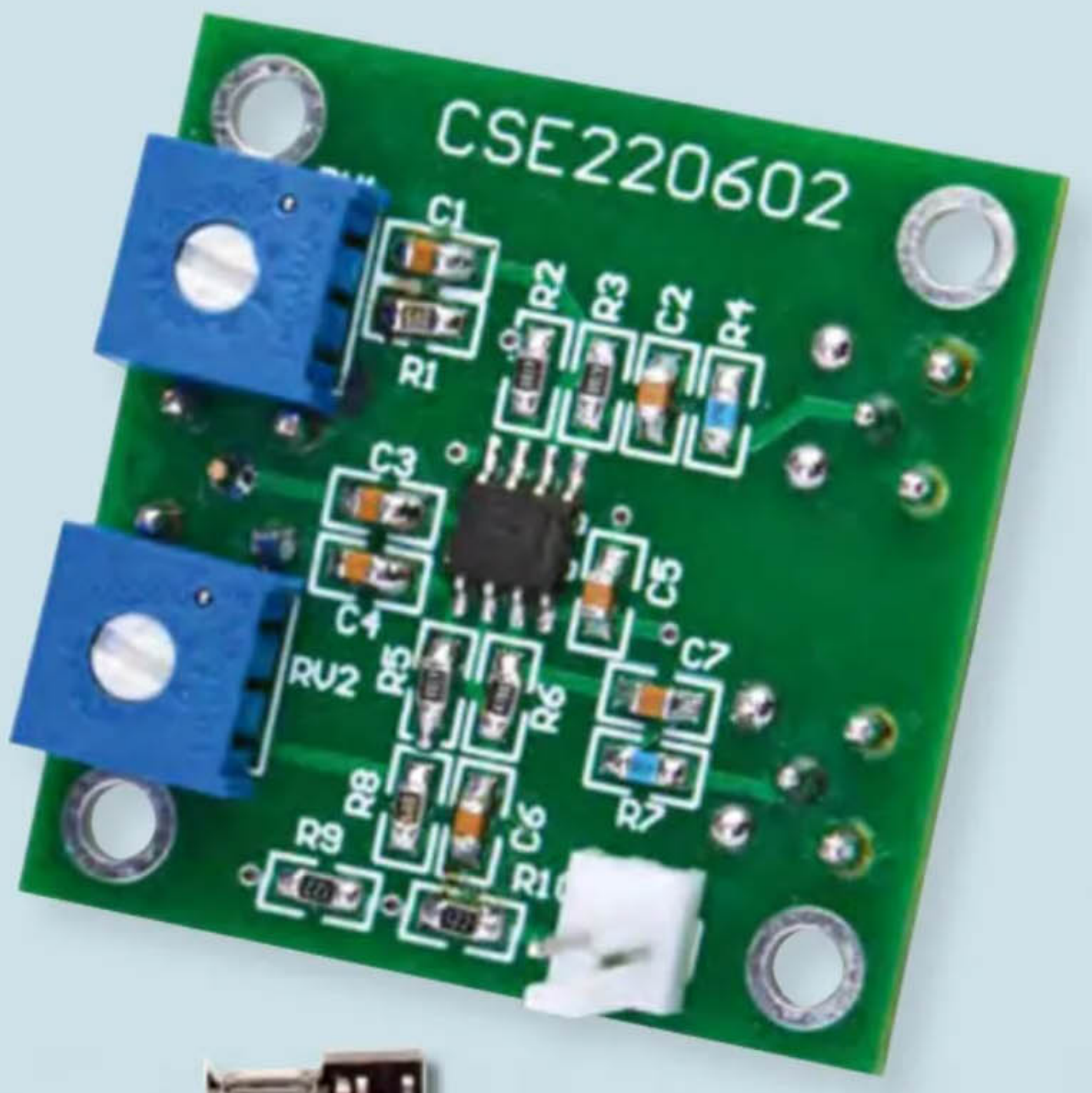
Web www.electronpublishing.com

For editorial contact details see page 7.

Next Month – in the May issue

Dual RF Amplifier for Signal Generators

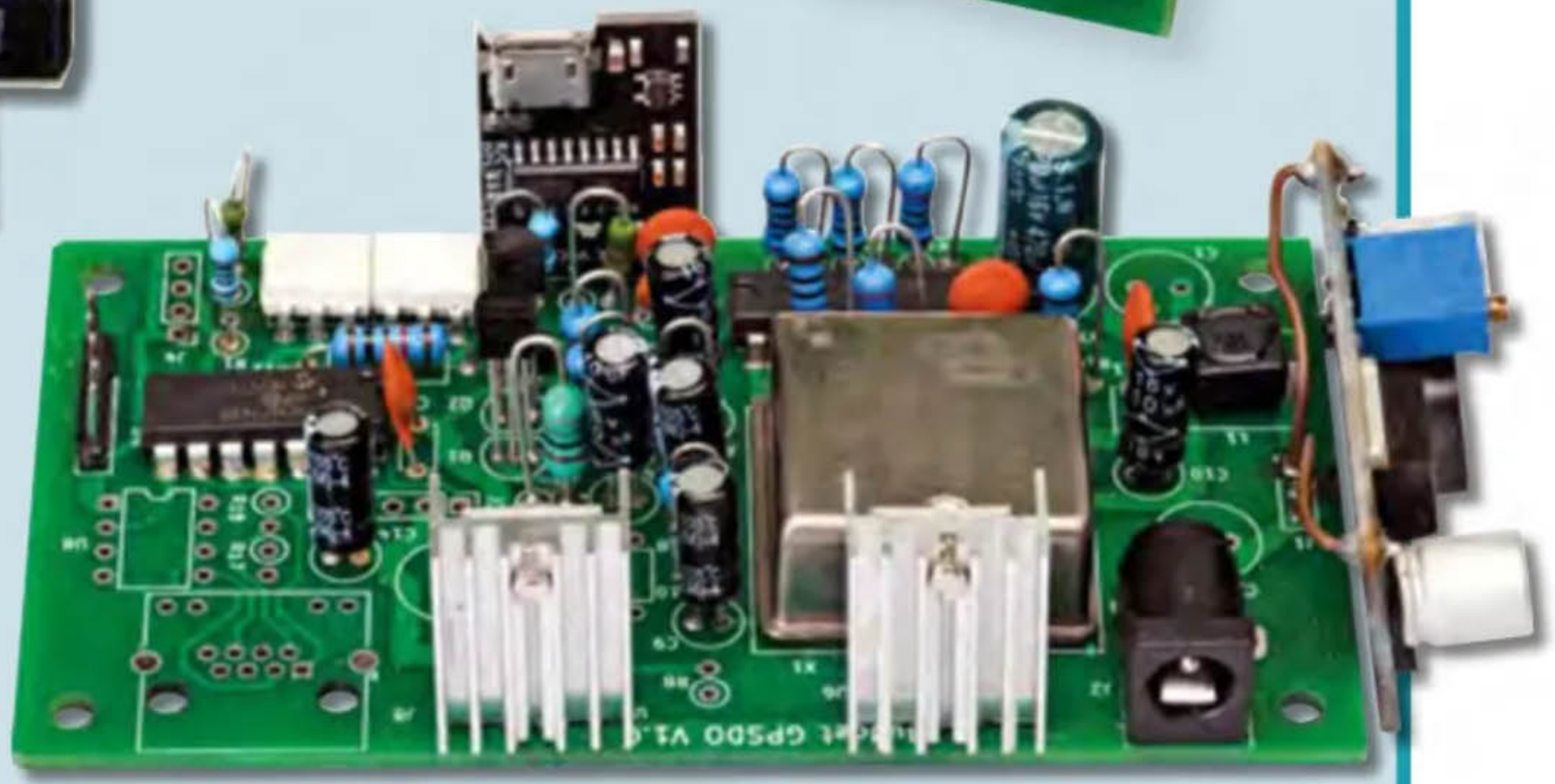
This small RF amplifier has two outputs with individually selectable gains. This makes it suitable to add to a signal generator to provide a higher output level, or for better drive strength, or 'fanning it out' to multiple other pieces of equipment and more.



Gsens
UVM30A

GPS-Disciplined Oscillator

The GPS-Disciplined Oscillator (GPSDO) is built almost entirely in software, so it only requires a PIC, an oven-conditioned crystal oscillator and a few other supporting parts. It provides an extremely accurate 10MHz signal with an error in the parts per billion range.



UVM-30A Module Ultraviolet Light Sensor

Learn how to use this ultraviolet (UV) light-sensing 'breakout' module. It detects the intensity of UV solar radiation and hence the degree of protection you may need to prevent skin damage. If you connect it to an Arduino or other microcontroller, it can even indicate the current 'UV Index'.

PLUS!

All your favourite regular columns from *Cool Beans* and *Circuit Surgery*, to *MitchElectronics*, *Teach-In*, *Techno Talk* and *Net Work*.

On sale 4 April 2024

Content may be subject to change

WELCOME TO JPG ELECTRONICS
Selling Electronics in Chesterfield for 29 Years

Open Monday to Friday 9am to 5:30pm
And Saturday 9:30am to 5pm

- Aerials, Satellite Dishes & LCD Brackets
- Audio Adaptors, Connectors & Leads
- BT, Broadband, Network & USB Leads
- Computer Memory, Hard Drives & Parts
- DJ Equipment, Lighting & Supplies
- Extensive Electronic Components
  - *ICs, Project Boxes, Relays & Resistors*
- Raspberry Pi & Arduino Products
- Replacement Laptop Power Supplies
- Batteries, Fuses, Glue, Tools & Lots more...

T: 01246 211 202
E: sales@jpgelectronics.com
JPG Electronics, Shaw's Row,
Old Road, Chesterfield, S40 2RB
W: www.jpgelectronics.com

JPG Electronics

Retail & Trade Welcome • Free Parking • Google St View Tour: S40 2RB

NEW subscriptions hotline!

Practical Electronics

We have changed the way we sell and renew subscriptions. We now use 'Select Publisher Services' for all print subscriptions – to start a new subscription or renew an existing one you have three choices:

1. Call our NEW print subscription hotline: 01202 087631, or email: pesubs@selectps.com
2. Visit our shop at: www.electronpublishing.com
3. Send a cheque (payable to: 'Practical Electronics') with your details to:
   Practical Electronics Subscriptions, PO Box 6337,
   Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 01202 880299 or visit: www.electronpublishing.com

Published on approximately the first Thursday of each month by Electron Publishing Limited, 1 Buckingham Road, Brighton, East Sussex BN1 3RA. Printed in England by Acorn Web Offset Ltd., Normanton WF6 1TW. Distributed by Seymour, 86 Newman St., London W1T 3EX. Subscriptions UK: £33.99 (6 months); £59.99 (12 months); £114.99 (2 years). EUROPE: airmail service, £38.99 (6 months); £69.99 (12 months); £129.99 (2 years). REST OF THE WORLD: airmail service, £46.99 (6 months); £84.99 (12 months); £164.99 (2 years). Payments payable to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom. Email: pesubs@selectps.com. PRACTICAL ELECTRONICS is sold subject to the following conditions, namely that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price shown on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter whatsoever.

Did you know our online shop now sells the current issue of *PE* for £5.99 inc. p&p?

Practical Electronics
The UK's premier electronics and computing maker magazine
Circuit Surgery
LTspice 24 update and Frequency Response Analysis
Teach-In 2024
Digital input and output with the ESP32 SoC microcontroller
Audio Out
Designing discrete buffers
Basic RF Signal Generator
'Swiss Army Knife' Automated Test Bench
WIN! Microchip Explorer 8 Development Kit WIN!
Teach-In 2024 Learn to use the ESP32
Silicon Chirp Build your own pet cricket
PLUS!
Techno Talk – Wait! What? Really?
Cool Beans – Arduino: expanding the 7-segment display
Net Work – Label printers, smart home devices and SMRs
Apr 2024 £5.99
www.electronpublishing.com
@practicalelec
practicalelectronics
Mar 2024 £5.99

You read that right! We now sell the current issue of your favourite electronics magazine for exactly the same price as in the High Street, but we deliver it straight to your door – and for UK addresses we pay the postage. No need to journey into town to queue outside the newsagent. Just go to our website, set up an account in 30 seconds, order your magazine and we'll do the rest.

www.electronpublishing.com

PEAK®
electronic design ltd

Use the code PE10
to claim your 10% discount!
(Offer to end soon)

Just imagine if you could:

Test almost any component
Identify unmarked/unknown components
Identify the pinout of components
Measure the parameters of components
Match components
Wind your own inductors
Fine tune your trimmer capacitors
Identify ideal operating points
Plot full datasheet-style charts
Identify faulty or degraded components
Identify alternatives

With Peak, *YOU* CAN.

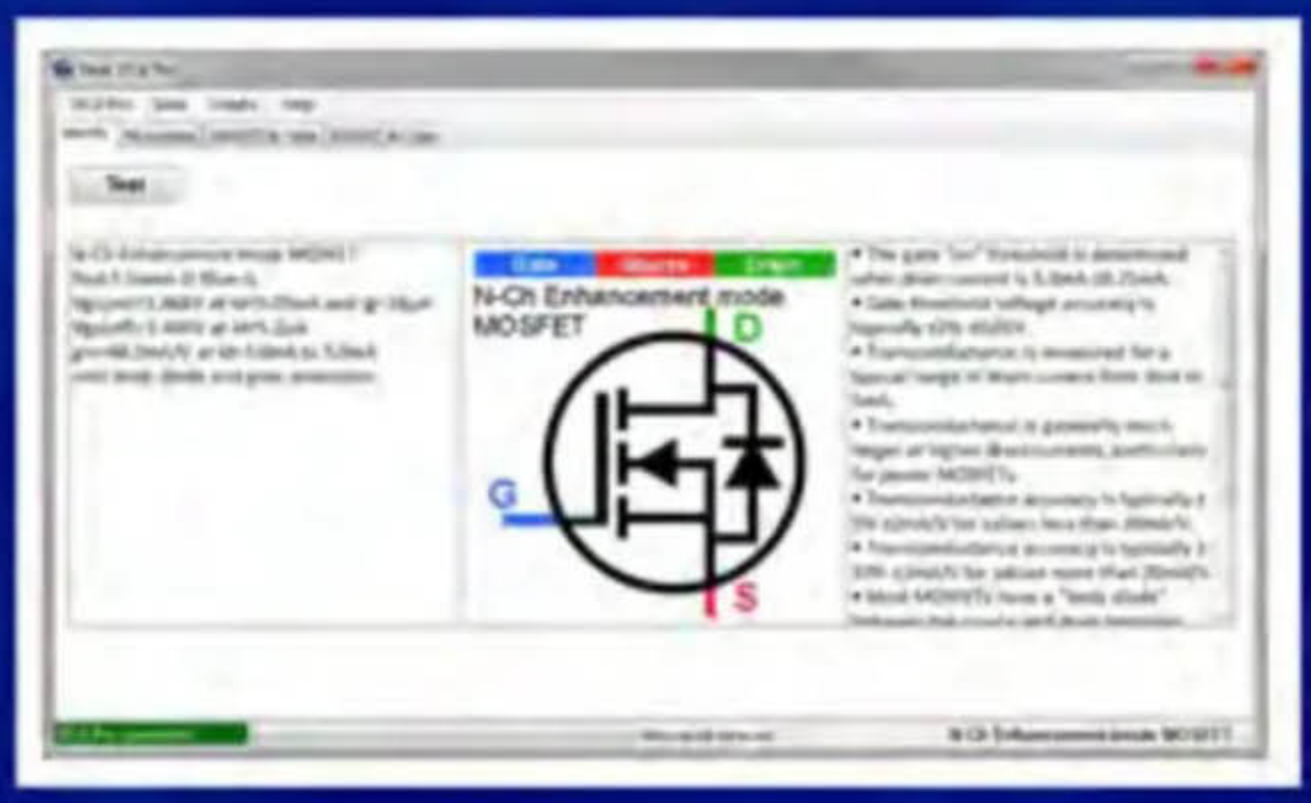
atlas ESR70 gold
Cap = 21.40uF
ESR = 0.78Ω
ESR, low resistance and capacitance meter

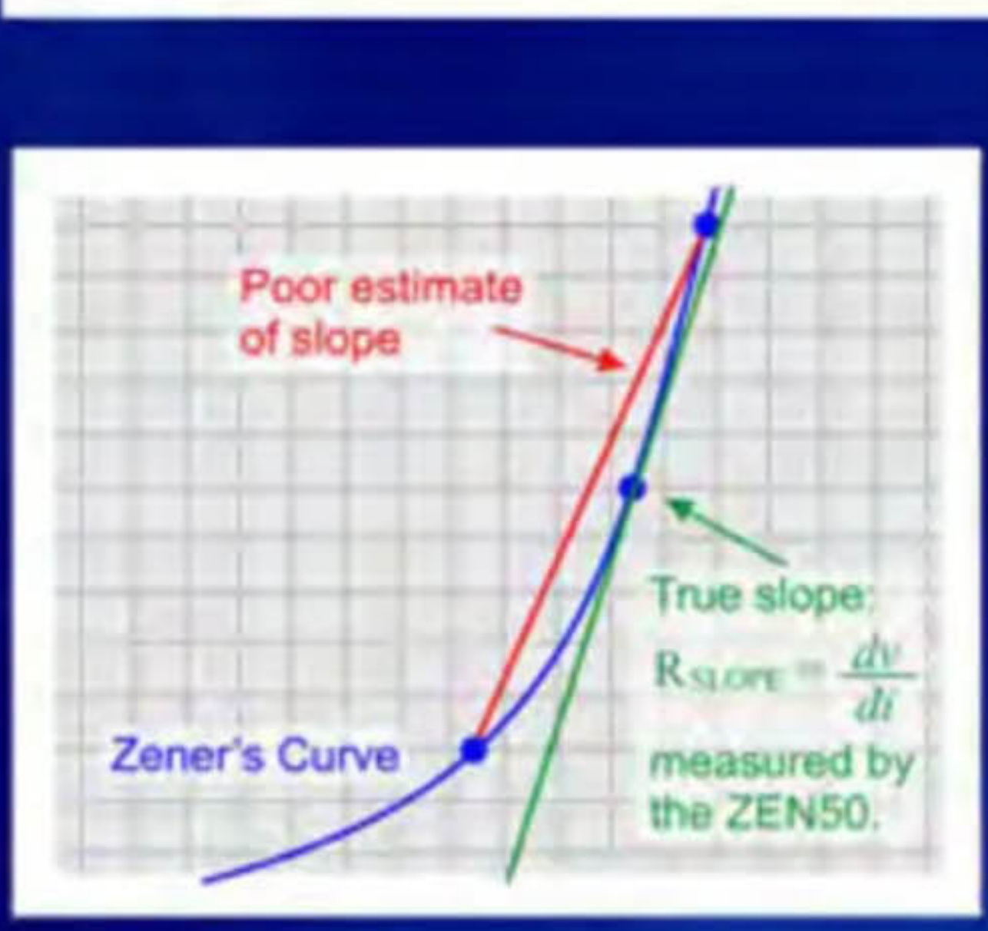
atlas LCR45
LCR meter with LCR impedance measurement

atlas DCA75 Pro
advanced semiconductor component analyser with USB and graphics

atlas DCA55
RED GREEN BLUE
Coll Base Emit
semiconductor component analyser

atlas ZEN50
Vz=46.49V @ 15mA
Slope R=16Ω
Zener diode analyser


N-Ch Enhancement mode MOSFET



Get guidance from the actual designers and manufacturers here in the heart of Derbyshire, UK.

UK designed and manufactured

Tel. 01298 70012 for genuine advice
order: www.peakelec.co.uk